BANKING MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

KAVITHA R	**240701244**
KAROLIN AUXILIA J	**240701228**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“BANKING MANAGEMENT SYSTEM”** is the bonafide work of **“KAVITHA R, KAROLIN AUXILIA J”** who carried out the project work under my supervision.

SIGNATURE

Mrs.B.Deepa

ASSISTANT PROFESSOR (SG)

Dept. of Computer Science and Engg,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

In our society, the banking sector plays a vital role in managing financial transactions efficiently. Even though many large national and international banks operate on a wider scale, small and local banks often lack an organized and automated system for managing customer accounts and transactions. To address this drawback, our team developed a **Banking Management System** to help local banks and financial institutions maintain their data and operations efficiently.The main objective of this project is to simplify and automate core banking functions such as **creating new accounts, depositing and withdrawing money, and viewing account details**. This system ensures accuracy, and ease of access to customer information, thereby improving the efficiency of banking operations. It also enables local banks to offer quick and reliable services, helping them stay competitive in today's fast-paced digital era.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Mrs.B.Deepa ,** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. KAVITHA R

2. KAROLIN AUXILIA J

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	**3**
1	**INTRODUCTION**	**7**
	1.1 INTRODUCTION	7
	1.2 SCOPE OF THE WORK	7
	1.3 PROBLEM STATEMENT	7
	1.4 AIM AND OBJECTIVES OF THE PROJECT	8
2	**SYSTEM SPECIFICATIONS**	**8**
	2.1 HARDWARE SPECIFICATIONS	8
	2.2 SOFTWARE SPECIFICATIONS	8
3	**MODULE DESCRIPTION**	**9**
4	**CODING**	**10**
5	**SCREENSHOTS**	**16**
6	**CONCLUSION AND FUTURE ENHANCEMENT**	**19**
7	**REFERENCES**	**20**

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	ACCOUNT CREATION	16
5.2	ACCOUNT CREATED	16
5.3	AMOUNT DEPOSITED	17
5.4	ACCOUNT AFTER DEPOSITED	17
5.5	AMOUNT WITHDRAWN	18
5.6	ACCOUNT AFTER WITHDRAWN	18

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Banking Management System is a computerized application developed to manage and automate various banking operations efficiently. It helps in maintaining accurate records of customers, their accounts, and transactions such as deposits, withdrawals, and fund transfers. The system reduces manual work, minimizes errors, and ensures secure handling of financial data. By providing a user-friendly interface and reliable database connectivity, the project aims to improve the overall efficiency and accuracy of banking services.

1.2 SCOPE OF THE WORK

The Banking Management System helps customers and bank staff manage accounts efficiently. It allows users to create accounts, deposit and withdraw money, and view account details easily. The system improves accuracy, reduces manual work, and ensures secure handling of transactions for local banks.

1.3 PROBLEM STATEMENT

Traditional banking systems rely heavily on manual record-keeping and paper-based transactions, which are time-consuming and prone to errors. Managing customer information, processing transactions, and maintaining account details manually often leads to data inconsistency and inefficiency. There is a need for a computerized system that can securely store data, automate banking operations, and provide quick access to information.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main aim of this project is to develop a system that helps banks manage customer accounts and transactions efficiently. The Banking Management System simplifies key operations such as creating accounts, depositing and withdrawing money, and viewing account details. It ensures accuracy, security, reducing manual effort and helping local banks provide better and more reliable services to their customers.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front – End	:	Java(Swing/AWT)
Back - End	:	MySql
Language	:	Java,SQL
Tools required	:	VS code,Mysql Workbench
Database Driver	:	MySql Connector/J

CHAPTER 3

MODULE DESCRIPTION

This application mainly consists of three modules. When the program runs, it allows the user to create a new account, perform transactions, and view account details. The modules are as follows:

1. Account Management Module

This module allows the user to create a new account by entering details such as name, account number, and initial deposit. It securely stores all customer information in the system.

2. Transaction Module

This module enables users to perform basic banking operations such as depositing and withdrawing money. It ensures secure and accurate updating of account balances.

3. Account View Module

This module allows users to view their account details, including account number, holder name, and current balance. It helps customers easily track and verify their financial information.

CHAPTER 4

SAMPLE CODING

Sample coding depicts the display code that retrieves data from the database and shows it to the user upon request.It fetches account details from the MySQL database and displays them in the console

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

public class BankingGUI extends JFrame {

private AccountDAO dao = new AccountDAO();

public BankingGUI() {

setTitle("Banking Management System");

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setSize(500, 400);

setLocationRelativeTo(null);

initUI();

}

private void initUI() {

JTabbedPane tabs = new JTabbedPane();

// Create Account panel
```

```
JPanel createPanel = new JPanel(new GridLayout(5,2,8,8));
JTextField accField = new JTextField();
JTextField nameField = new JTextField();
JComboBox<String> typeBox = new JComboBox<>(new String[]{"Savings","Checking"});
JTextField balField = new JTextField("0.00");
JButton createBtn = new JButton("Create Account");
createPanel.add(new JLabel("Account Number:")); createPanel.add(accField);
createPanel.add(new JLabel("Name:")); createPanel.add(nameField);
createPanel.add(new JLabel("Account Type:")); createPanel.add(typeBox);
createPanel.add(new JLabel("Initial Balance:")); createPanel.add(balField);
createPanel.add(new JLabel()); createPanel.add(createBtn);
createBtn.addActionListener(e -> {
try {
int acc = Integer.parseInt(accField.getText().trim());
String name = nameField.getText().trim();
String type = (String)typeBox.getSelectedItem();
double bal = Double.parseDouble(balField.getText().trim());
boolean ok = dao.createAccount(acc, name, type, bal);
JOptionPane.showMessageDialog(this, ok ? "Account created." : "Failed to create account.");
```

```
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(this, "Invalid number format.");
}
});
// Deposit / Withdraw panel
JPanel transPanel = new JPanel(new GridLayout(4,2,8,8));
JTextField tAccField = new JTextField();
JTextField amtField = new JTextField();
JButton depositBtn = new JButton("Deposit");
JButton withdrawBtn = new JButton("Withdraw");
transPanel.add(new JLabel("Account Number:")); transPanel.add(tAccField);
transPanel.add(new JLabel("Amount:")); transPanel.add(amtField);
transPanel.add(depositBtn); transPanel.add(withdrawBtn);
transPanel.add(new JLabel()); transPanel.add(new JLabel());
depositBtn.addActionListener(e -> {
try {
int acc = Integer.parseInt(tAccField.getText().trim());
double a = Double.parseDouble(amtField.getText().trim());
boolean ok = dao.deposit(acc, a);
JOptionPane.showMessageDialog(this, ok ? "Deposit successful." : "Deposit failed.");
```

```
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(this, "Invalid number format.");
}
});
withdrawBtn.addActionListener(e -> {
try {
int acc = Integer.parseInt(tAccField.getText().trim());
double a = Double.parseDouble(amtField.getText().trim());
boolean ok = dao.withdraw(acc, a);
JOptionPane.showMessageDialog(this, ok ? "Withdrawal successful." : "Withdrawal failed (insufficient funds or account missing)." );
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(this, "Invalid number format.");
}
});
// View Account panel
JPanel viewPanel = new JPanel(new BorderLayout(8,8));
JPanel top = new JPanel(new FlowLayout());
JTextField vAccField = new JTextField(10);
JButton viewBtn = new JButton("View Account");
JTextArea resultArea = new JTextArea();
```

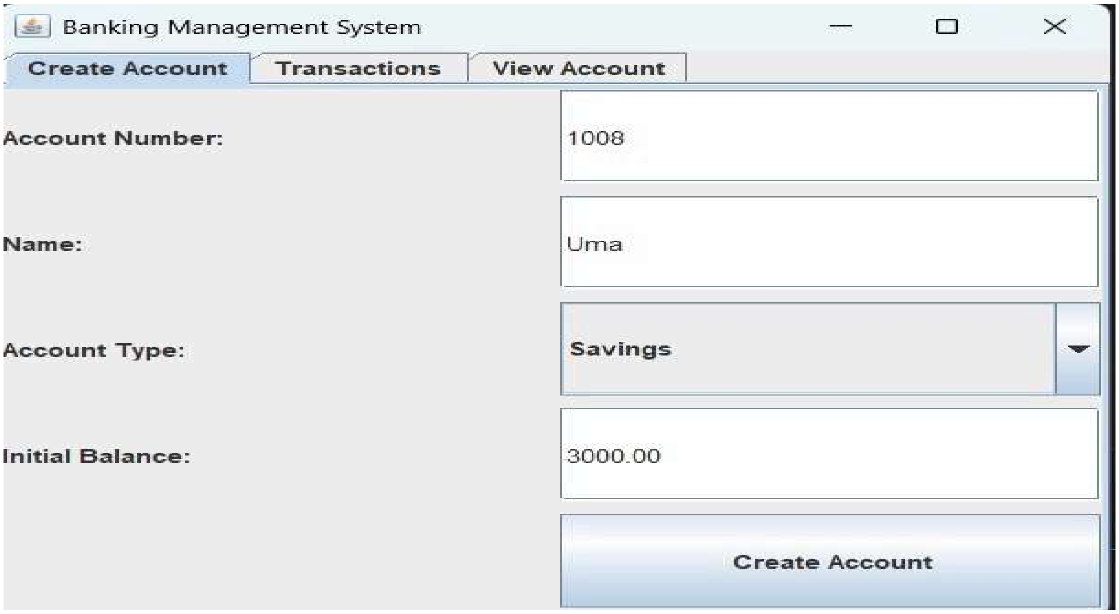
```
resultArea.setEditable(false);
top.add(new JLabel("Account Number:")); top.add(vAccField); top.add(viewBtn);
viewPanel.add(top, BorderLayout.NORTH);
viewPanel.add(new JScrollPane(resultArea), BorderLayout.CENTER);
viewBtn.addActionListener(e -> {
try {
int acc = Integer.parseInt(vAccField.getText().trim());
String res = dao.getAccountInfo(acc);
resultArea.setText(res);
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(this, "Invalid account number.");
}
});
tabs.addTab("Create Account", createPanel);
tabs.addTab("Transactions", transPanel);
tabs.addTab("View Account", viewPanel);
add(tabs, BorderLayout.CENTER)
```

Sample 1

```
CREATE DATABASE IF NOT EXISTS bankdb;

USE bankdb;

CREATE TABLE IF NOT EXISTS Accounts (

AccountNumber INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

AccountType VARCHAR(20) NOT NULL,

Balance DECIMAL(12,2) NOT NULL DEFAULT 0.00,

CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);

-- Sample data

INSERT INTO Accounts (AccountNumber, Name, AccountType, Balance)
VALUES

(1001, 'Alice', 'Savings', 1500.00),

(1002, 'Bob', 'Checking', 800.00);
```

CHAPTER 5

SCREEN SHOTS



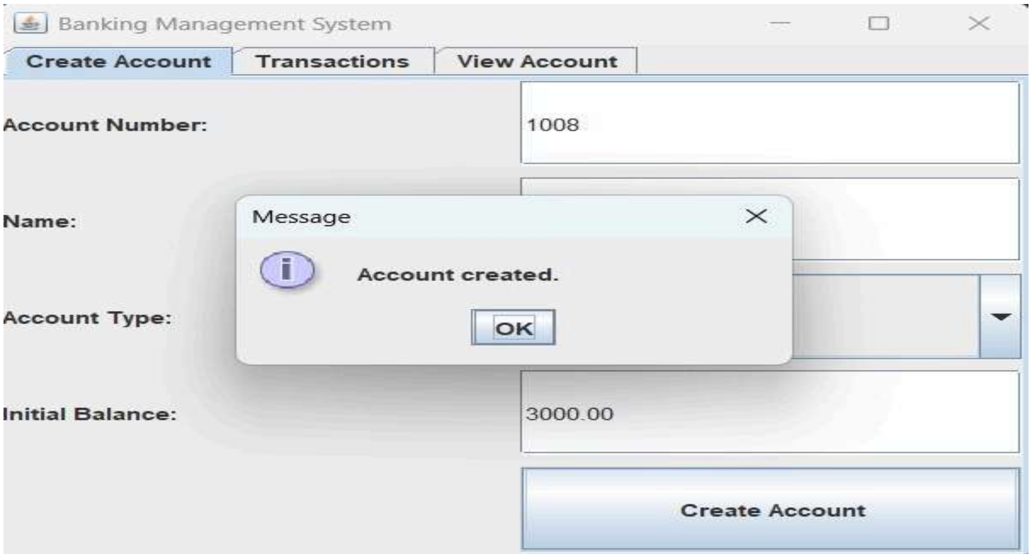
Account Number: 1008

Name: Uma

Account Type: Savings

Initial Balance: 3000.00

Fig 5.1 Account creation



Account Number: 1008

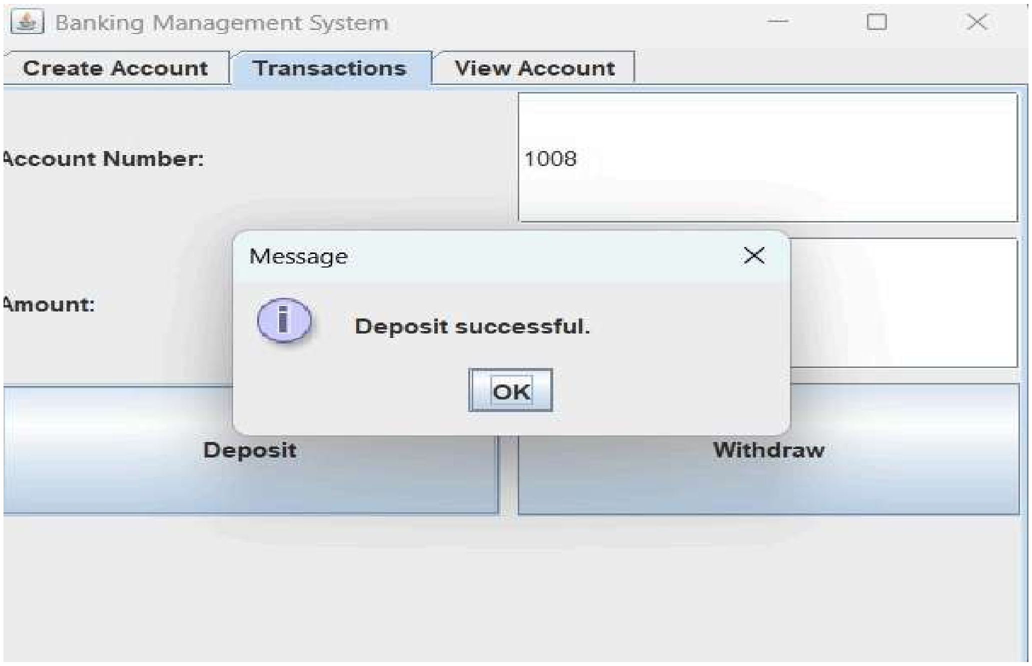
Name:

Account Type:

Initial Balance: 3000.00

Account created.

Fig 5.2 Account created

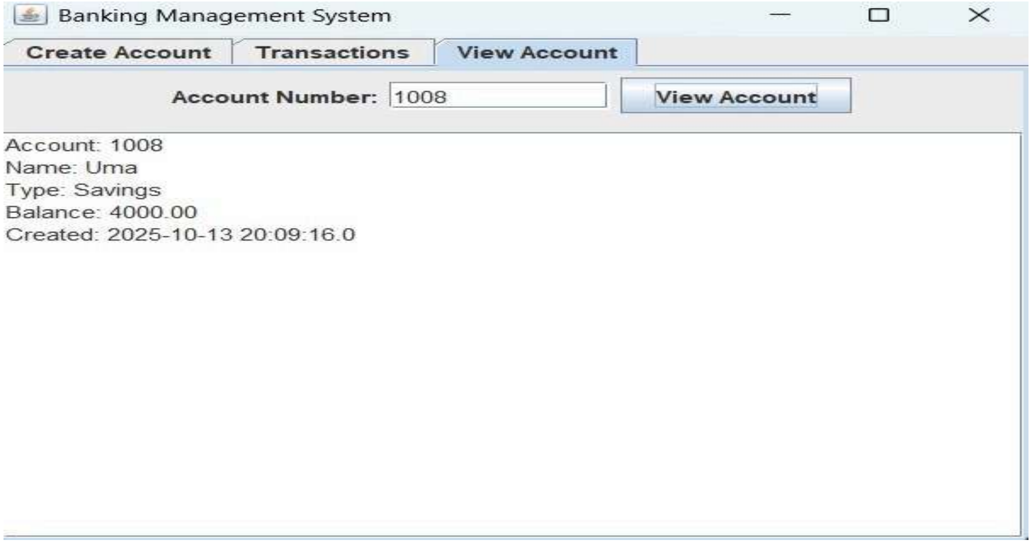


Account Number: 1008

Amount:

Deposit successful.

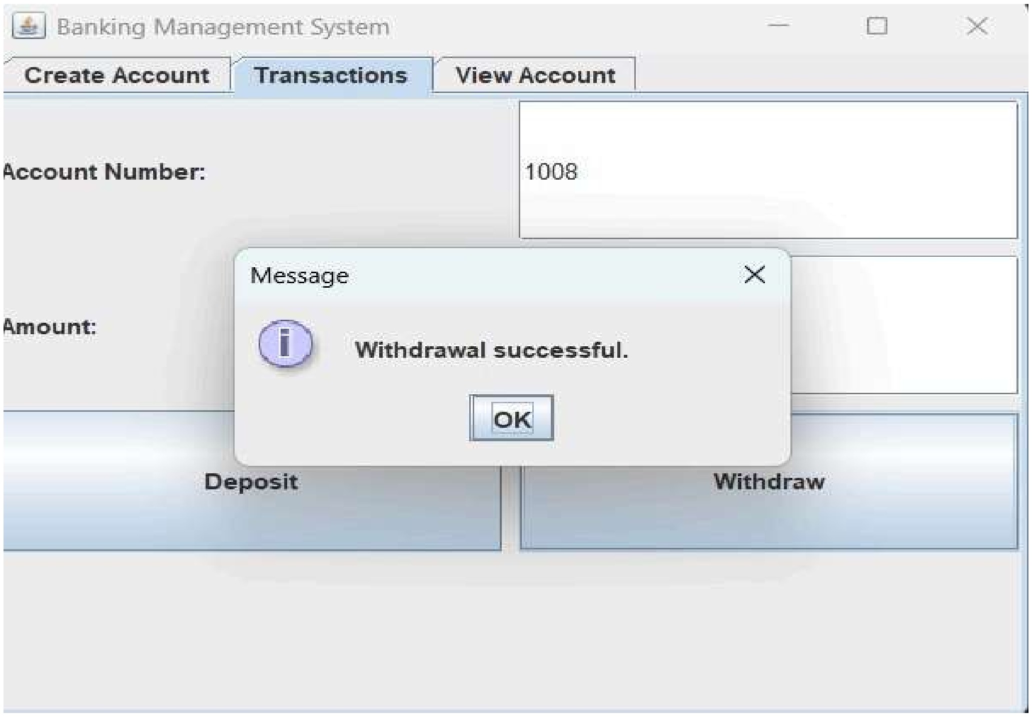
Fig 5.3 Amount deposited



Account Number: 1008

```
Account: 1008
Name: Uma
Type: Savings
Balance: 4000.00
Created: 2025-10-13 20:09:16.0
```

Fig 5.4 Account after deposited



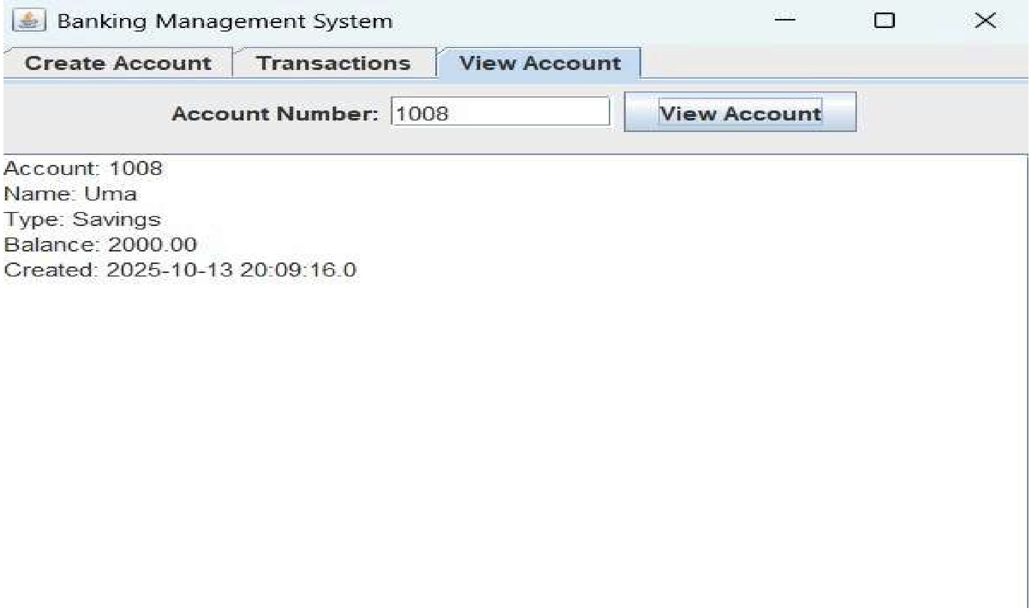
Account Number: 1008

Amount:

Message

Withdrawal successful.

Fig 5.5 Amount withdrawn



Account Number: 1008

```
Account: 1008
Name: Uma
Type: Savings
Balance: 2000.00
Created: 2025-10-13 20:09:16.0
```

Fig 5.6 Account after withdrawn

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

In this project, we have developed a Banking Management System that helps users perform basic banking operations such as creating new accounts, depositing and withdrawing money, and viewing account details. The system makes banking activities easier to manage and reduces the need for manual record-keeping. It provides a simple way to handle account information and transactions efficiently.

In the future, this system can be enhanced by adding features such as interest calculation, loan processing, and automatic report generation. These additions would make the system more useful and suitable for real-world banking applications.

REFERENCES

1. https://www.w3schools.com/java/
2. https://www.w3schools.com/sql/ w3schools.com
3. https://www.geeksforgeeks.org/java/
4. https://www.w3schools.com/sql/
5. https://dev.mysql.com/doc/
6. https://code.visualstudio.com/doc
7. https://www.academia.edu/Documents/in/Bank_Management
8. https://www.jetir.org/papers/JETIR2305489.pdf
9. https://www.jetir.org/papers/JETIR1809441.pdf
10. https://ijcrt.org/papers/IJCRT2404083.pdf