

CHAPTER 1

INTRODUCTION

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

A Bank management system is designed to handle all the primary information required for maintaining a person's account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash and other core banking features from his account. It also enables customer to view reports of all accounts present, calculate monthly statements of customer account which include monthly statement of any month.

The bank management system is an essential component of modern banking, as it helps to ensure the smooth functioning of financial institutions. It provides a comprehensive solution for managing various banking processes and operations, making it easier for banks to offer their services to customers. Here are some of the key features and benefits of a bank management system:

1. **Account Management:** The bank management system enables financial institutions to manage customer accounts efficiently. It allows banks to open and close accounts, perform account maintenance tasks, and manage account transactions.
2. **Transaction Processing:** The system enables banks to process various types of transactions, such as deposits, withdrawals, and transfers. It ensures that transactions are completed accurately and efficiently.

3. Loan Management: The bank management system enables financial institutions to manage loan applications, approvals, and disbursements. It also helps to monitor loan repayments and ensures that loans are managed effectively.
4. Customer Relationship Management: The system enables banks to manage their relationships with customers effectively. It provides a platform for customer support, complaint management, and feedback management.
5. Security and Compliance: The bank management system ensures that all transactions are secure and comply with the regulations and policies of the financial institution. It also provides tools for managing fraud and security risks.
6. Reporting and Analytics: The system provides banks with real-time data on their operations, allowing them to make informed decisions. It also provides tools for generating reports and analyzing data to identify trends and areas for improvement.

Separate database is maintained to handle all the details required for the correct statement calculation and generation. This project intends to introduce more user friendliness in the various activities such as record, updating, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number with the password of that customer. Similarly, record maintenance and updating can also be accomplished by using the account number and password with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

1.1 BRIEF INTRODUCTION

A Bank management system is designed to handle all the primary information required for maintaining a person's account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash and other core banking features from his account. It also enables customer to view reports of all accounts present, calculate monthly statements of customer account which include monthly statement of any month.

Separate database is maintained to handle all the details required for the correct statement calculation and generation. This project intends to introduce more user friendliness in the various activities such as record, updating, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number with the password of that customer. Similarly, record maintenance and updating can also be accomplished by using the account number and password with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to- date.

1.2 MOTIVATION

The motivation for creating a bank management system can be attributed to several factors. Firstly, the increasing demand for banking services and the need to provide fast and efficient services to customers has led to the adoption of technology in banking operations. A bank management system provides a platform for financial institutions to manage their operations efficiently, automate various processes, and offer a wide range of services to customers.

Secondly, the need for security and compliance has also motivated the development of bank management systems. Financial institutions are required to comply with regulations and policies to ensure the security of their operations and customers' data. A bank management system provides tools for managing fraud and security risks, monitoring transactions, and ensuring compliance with regulations.

Thirdly, the need for real-time data and analytics has also driven the development of bank management systems. Financial institutions require up-to-date data on their operations to make informed decisions, identify trends, and areas for improvement. A bank management system provides real-time data, reporting, and analytics tools, enabling financial institutions to make data-driven decisions.

Finally, the need to improve customer experience and engagement has also motivated the development of bank management systems. Customers expect fast, efficient, and personalized services from their financial institutions. A bank management system provides a platform for managing customer accounts, processing transactions, and providing online and mobile banking services, enabling financial institutions to offer a wide range of services to customers.

In conclusion, the motivation for creating a bank management system can be attributed to the increasing demand for banking services, the need for security and compliance, the need for real-time data and analytics, and the need to improve customer experience and engagement. A bank management system provides a comprehensive solution for managing various banking operations, enabling financial institutions to stay competitive and meet the evolving needs of their customers.

1.3 SCOPE

The scope of a bank management system is vast and can encompass several areas of a bank's operations. Some of the key areas where a bank management system can have an impact include:

1. Customer Management: A bank management system can enable banks to manage customer data more efficiently, including account information, transaction history, and personal details. This can help to provide a better customer experience and improve customer satisfaction.
2. Account Management: A bank management system can enable banks to manage customer accounts more efficiently, including opening and closing accounts, processing transactions, and maintaining account balances. This can help to reduce errors and improve the accuracy of account data.
3. Reporting: A bank management system can provide banks with real-time data and analytics on various aspects of their operations, including customer behaviour, transaction volume, and profitability. This can help to improve decision-making and optimize business processes.

1.4 PROBLEM STATEMENT

A bank management system website is needed to streamline the banking operations of a financial institution. The website should provide a user-friendly interface that enables customers to access various banking services such as account management, funds transfer, bill payments, and loan applications. The website should also provide the bank employees with tools for managing customer accounts, tracking transactions, and generating reports.

The bank management system website should have the following features:

1. User registration and login: The website should allow users to create an account and login securely. Users should be able to reset their password in case they forget it.
2. Account management: The website should allow customers to view their account balances, transaction history, and other details related to their account. Customers should be able to make account-related changes, such as updating their contact information.
3. Funds transfer: The website should enable customers to transfer funds between their accounts or to other accounts within the same bank. Customers should also be able to transfer funds to accounts at other banks.
4. Card Details: The website should enable customers to view their card details and also provide to activate or deactivate cards.
5. Security features: The website should have strong security features such as encryption, firewalls, and user authentication to protect customer data and prevent unauthorized access.

Overall, the bank management system website should provide a seamless banking experience for customers while enabling bank employees to manage customer accounts efficiently.

1.5 PROPOSED SYSTEM

The application will be extremely beneficial for the Customers intending to use and operate their bank account and will get various benefits in the field of management of accounts on clean and user-friendly platform. “Bank Management system”, is a website, which is especially generated and designed for the bank in order to enter the applicant information about his other bank account and can perform other function like currency change. It is user name, ID and password protected as well. Following are the major objectives behind the new proposed system:

- It creates a user-friendly environment, where a normal user can access through all the benefits of the system.
- The password is highly protected by hashing.
- It provides security from unauthorized access, only admin or authorized users are granted access to the system.
- It increases efficiency and saves the time.
- No any danger and obstacles from external entities.
- Easy access of saved data inside the system.
- Complex Banking operations and Transaction operations are efficiently handled by the application
- It has ease of use along with complete reference
- It is highly secured and less time consuming; hence time wastage can be avoided
- Up to date records of the customers are maintained by the authority.

1.6 LIMITATIONS

1. Less security of customer and bank information.
2. Require more physical work and manpower.
3. All the manual entry and editing will take more time.
4. No level of clearance for the different levels of employees.
5. Safety of paper documents from the disaster.
6. No backup of the information.
7. Existing online systems are not user friendly

CHAPTER 2

LITERATURE SURVEY

1. "Design and Implementation of an Online Banking System" by Oluwaseun Akinwale and Adebayo Adesina, published in the International Journal of Computer Applications in 2015. This paper presents the design and implementation of an online banking system using PHP, MySQL, and HTML. The system includes features such as user registration, account management, funds transfer, bill payments, and loan applications.
 2. "An Overview of Online Banking and its Security Issues" by Zahidur Rahman and Md. Farhanul Hossain, published in the International Journal of Computer Applications in 2013. This paper provides an overview of online banking and discusses the security issues associated with it. The authors present various security measures such as two-factor authentication, encryption, and firewalls that can be used to protect online banking systems.
 3. "Development of a Web-Based Banking System" by Abimbola O. Adubi and Olufunke O. Adubi, published in the International Journal of Computer Science and Information Technology in 2015. This paper describes the development of a web-based banking system using PHP, MySQL, and HTML. The system includes features such as user registration, account management, funds transfer, and bill payments.
 4. "A Comparative Study of Traditional Banking System and Online Banking System" by Shamsul Arefin, published in the International Journal of Science and Research in 2015. This paper presents a comparative study of traditional banking systems and online banking systems. The author discusses the advantages and disadvantages of both systems and concludes that online banking is more convenient and efficient.
 5. "A Secure Online Banking System Using Biometrics" by Tariq M. Yousef, published in the International Journal of Computer Applications in 2012. This paper presents the design and implementation of a secure online banking system using biometric authentication. The system includes features such as user registration, account management, funds transfer, bill payments, and loan applications.
 6. "Design and Implementation of an Online Banking System with SMS Notification" by Joseph Adeyemi Adebiyi and Segun O. Olatinwo, published in the Journal of Information Engineering and Applications in 2015. This paper presents the design and
-

implementation of an online banking system with SMS notification using PHP, MySQL, and HTML. The system includes features such as user registration, account management, funds transfer, bill payments, and loan applications.

7. "Implementation of a Secure Online Banking System" by K. D. Joshi and P. S. Deshpande, published in the International Journal of Computer Science and Network Security in 2013. This paper presents the implementation of a secure online banking system using Java, JSP, and MySQL. The system includes features such as user registration, account management, funds transfer, bill payments, and loan applications. The authors also discuss the security measures used in the system, such as SSL encryption and digital signatures.

Overall, these papers provide a comprehensive overview of online banking systems and the various technologies used in their development. They also discuss the security issues associated with online banking and present various measures that can be used to protect these systems.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

The system requirements for a bank management system using NODE.JS , HTML, and CSS are as follows:

Hardware Requirements:

1. A computer system with a minimum of 2 GHz Processor.
2. Smartphones with Android 4.4 and higher or iOS 9.0 and higher.
3. Processor: Intel Core i3 or equivalent
4. RAM: 4GB or more
5. Hard disk: 500GB or more
6. Network card: Ethernet or Wi-Fi

Software Requirements:

1. An internet connection.
2. Operating System: Windows 7 or later, macOS, or Linux
3. Terminal or Command Prompt to run the servers.
4. Database: MySQL
5. Text Editor: Visual Studio Code, Sublime Text, or any other text editor .
6. Browser: Chrome, Firefox, Safari, or Edge.

Development Tools:

1. Git: For version control
2. Composer: For package management
3. Node.js: For front-end development and building tools
4. CSS pre-processor: Sass, Less, or Stylus

Functional Requirements:

1. User Registration and Login: Users should be able to register for an account and log in to the system using their credentials.
2. Forgot Password: Users should be able to reset their password if they forgotten it.
3. Dashboard : The system should provide a dashboard showing monthly credited and debited money, account information and recent transactions.
4. Transfer : Users should be able to transfer money from their account to others account through their account number
5. Transaction History: Users should be able to view their transaction history to track their usage.
6. Card Details: Users should be able to view their card details. Activate and Deactivate their cards.

Non-functional Requirements:

1. User Interface: The user interface should be intuitive, user-friendly, and visually appealing.
2. Security: The system should provide robust security measures, including encryption, firewalls, and a secure sockets layer (SSL) to protect user data.
2. Performance: The system should be highly responsive and have a fast loading time to ensure a seamless user experience.
3. Scalability: The system should be scalable and able to handle a large number of users and transactions.
4. Compatibility: The system should be compatible with various browsers and operating systems, including Android, iOS, and Windows.

CHAPTER 4

SYSTEM ANALYSIS

System analysis for a bank management system involves identifying the requirements and functions of the system and designing a solution that meets these requirements. System analysis for a bank management system requires a thorough understanding of the bank's business processes, requirements, and technology infrastructure. It involves designing, developing, implementing, and maintaining a system that meets the needs of the bank and its customers, while also being scalable, secure, and reliable.

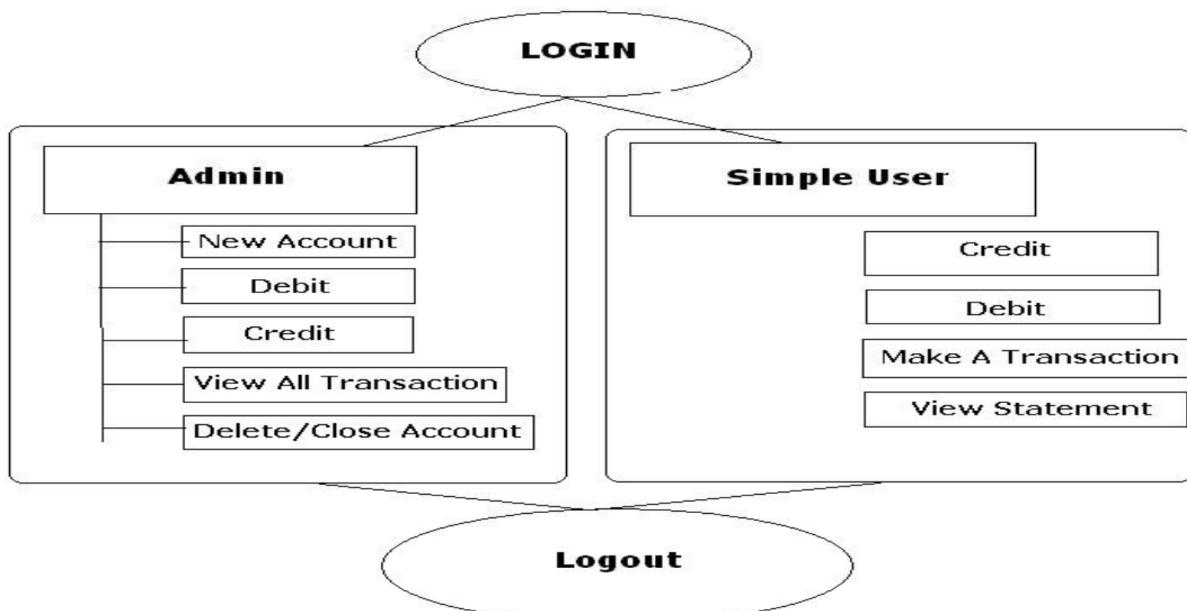


Figure 4.1 Use-Case Diagram

Field	Type	Null	Key	Default	Extra
accountNumber	bigint	YES	MUL	NULL	
cardName	varchar(80)	NO		NULL	
cardNumber	varchar(16)	NO	PRI	NULL	
cvv	int	NO		NULL	
issuedDate	date	NO		NULL	
expiryDate	date	NO		NULL	
cardStatus	varchar(12)	NO		NULL	

Figure 4.2 Class Diagram For Table :CARDS

Field	Type	Null	Key	Default	Extra
accountNumber	bigint	YES	MUL	NULL	
balance	float	NO		NULL	

Figure 4.3 Class Diagram For Table :DASHBOARD

Field	Type	Null	Key	Default	Extra
accountNumber	bigint	NO	PRI	NULL	
name	varchar(250)	YES		NULL	
contactNumber	varchar(20)	YES		NULL	
email	varchar(50)	YES		NULL	
password	varchar(50)	YES		NULL	
status	varchar(20)	YES		NULL	
role	varchar(20)	YES		NULL	

Figure 4.4 Class Diagram For Table :CUSTOMER

Field	Type	Null	Key	Default	Extra
transactionId	int	NO	PRI	NULL	auto_increment
accountNumber	bigint	YES	MUL	NULL	
transtype	varchar(250)	YES		NULL	
fromacc	varchar(250)	YES		NULL	
toacc	varchar(250)	YES		NULL	
amount	float	YES		NULL	
dateo	date	YES		NULL	
description	varchar(250)	YES		NULL	

Figure 4.5 Class Diagram For Table :TRANSACTION

CHAPTER 5

SYSTEM IMPLEMENTATION

1. Frontend Development: The frontend development will involve creating the user interface using HTML, CSS, and JavaScript. The user interface will allow users to input their information and interact with the system.
2. RestAPI Development: The RestAPI will be developed using Node, a javascript web application framework. The RestAPI will be responsible for receiving the user's input from the frontend and processing it.
3. Put Request: The frontend will make a Put request to the RestAPI endpoint with the user's input as the payload. The RestAPI will receive the request and process the payload.
4. RestAPI Processing: The RestAPI will process the user's input, which may include validation, data transformation, or calling other APIs or services.
5. Response: The RestAPI will generate a response to the frontend based on the user's input. The response may include data, status codes, or error messages.
6. Frontend Processing: The frontend will receive the RestAPI response and process it using JavaScript. The frontend may display the response to the user, update the user interface, or perform other actions based on the response.
7. Testing and Debugging: The system will be thoroughly tested and debugged to ensure that it meets the requirements and functions as expected. Testing may include unit testing, integration testing, and user acceptance testing.
8. Deployment: The system will be deployed to a production environment, which may include a web server or cloud-based platform. The deployment will involve configuring the system for scalability, security, and performance.
9. Monitoring and Maintenance: The system will be monitored and maintained to ensure that it continues to function properly and meets the needs of users. Maintenance may include bug fixes, upgrades, and performance optimization.
10. Documentation: The system will be documented to provide instructions for users and developers, including user manuals, API documentation, and code documentation. Documentation will help ensure that the system can be easily maintained and updated in the future.

NODE.JS Backend

1. Server.js

```
require('dotenv').config();
const http = require('http');
const app = require('./index');
const server = http.createServer(app);
server.listen(process.env.PORT);
```

2. Connection.js

```
const mysql = require('mysql');
require('dotenv').config();
var connection = mysql.createConnection ({
  port: process.env.DB_PORT,
  host: process.env.DB_HOST,
  user: process.env.DB_USERNAME,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME
});
connection.connect ((err)=>{
  if(!err){
    console.log("Connected");
  }
  else{
    console.log("sql connection error");
    console.log(err);
  }
});
module.exports = connection;
```

3. Post API ‘signup’

```
router.post('/signup',(req,res)=>{
  let user = req.body;
  console.log(req.body);
  query ="select accountNumber,name,email,password from customer where email=?"
  connection.query(query,[user.email],(err,results)=>{
    if(!err){
      if (results.length<=0){
        var accgen = getdig10();
        connection.query(query,[user.name,user.contactNumber,user.email,user.password],(er
        r,results)=>{
          if(err) throw err;
        });
        connection.query(query1,[],(err,results)=>{
          if(err) throw err;
        });
        connection.query(query3,[user.name],(err,results)=>{
```

```
        if(err) throw err;
    });
    connection.query(query4,[user.name],(err,results)=>{
        if(err) throw err;
    });
    connection.query(query2,[user.name],(er,results)=>{
        if(!err){
            return res.status(200).json({message:"Successfully registered"});
        }
        else{
            return res.status(500).json(err);
        }
    });
}
else{
    return res.status(400).json({message :"Email already Exists"});
}
}
else{
    return res.status(500).json(err);
}
});
});
```

4. Post API ‘login’

```
router.post('/adminlogin',(req,res)=>{
    const user = req.body;
    query ="select accountNumber,name,email,password,status,role from customer
where email=?";
    connection.query(query,[user.email],(err,results)=>{
        if(!err){
            if(results.length <=0 || results[0].password!= user.password ){
                return res.status(401).json({message: "Invalid Username or password"});
            }
            else if(results[0].role==='user'){
                return res.status(402).json({message:"Not an admin"});
            }
            else if(results[0].password == user.password){
                const response ={email :results[0]};
                const
accesToken=jwt.sign(response,process.env.ACCESS_TOKEN,{expiresIn:'8h'});
                return res.status(200).json({token:accesToken});
            }
            else{
                return res.status(400).json({message:"something went wrong"});
            }
        }
    });
});
```

```
});
```

5. Post API ‘dashboard’

```
router.post('/dashboard',(req,res)=>{
  const user=req.body;
  query1="SELECT * FROM customer JOIN transaction ON
customer.accountNumber = transaction.accountNumber JOIN cards ON
customer.accountNumber = cards.accountNumber JOIN dashboard ON
customer.accountNumber=dashboard.accountNumber WHERE customer.email = ?"
  connection.query(query1,[user.email],(err,results)=>{
    if(!err){
      return res.status(200).json(results);
    }
    else{
      return res.status(500).json(err);
    }
  })
});
```

6. Post API ‘transaction history’

```
router.post('/transactionHistory',(req,res)=>{
  const user=req.body;
  query1="SELECT * FROM customer JOIN transaction ON
customer.accountNumber = transaction.accountNumber where email=?"
  connection.query(query1,[user.email],(err,results)=>{
    if(!err){
      return res.status(200).json(results);
    }
    else{
      return res.status(500).json(err);
    }
  })
});
```

7. Post API ‘admin login’

```
router.post('/adminlogin',(req,res)=>{
  const user = req.body;
  query ="select accountNumber,name,email,password,status,role from customer
where email=?"
  connection.query(query,[user.email],(err,results)=>{
    if(!err){
      if(results.length <=0 || results[0].password!= user.password ){
        return res.status(401).json({message: "Invalid Username or password"});
      }
      else if(results[0].role==='user'){
        return res.status(402).json({message:"Not an admin"});
      }
    }
  })
});
```

```
        }
        else if(results[0].password == user.password){
            const response = {email :results[0]};
            const
            accesToken=jwt.sign(response,process.env.ACCESS_TOKEN,{expiresIn:'8h'});
            return res.status(200).json({token:accesToken});
        }
        else{
            return res.status(400).json({message:"something went wrong"});
        }
    });
});
```

8. Post API ‘admin dashboard’

```
router.post('/adminDash',(req,res)=>{
    const user = req.body;
    query ="SELECT * FROM customer JOIN dashboard ON customer.accountNumber
    = dashboard.accountNumber where role='user'";
    connection.query(query,[],(err,results)=>{
        if(!err){
            return res.status(200).json(results);
        }
        else{
            return res.status(500).json(err);
        }
    })
});
```

9. Post API ‘transfer’

```
router.post('/transfer',(req,res)=>{
    const user = req.body;
    query ="select * from customer where accountNumber=?"
    connection.query(query,[user.accountNumber],(err,results)=>{
        if(!err){
            if(results.length <=0 || results[0].accountNumber!= user.accountNumber ){
                return res.status(401).json({message: "Invalid Account Number "});
            }
            else if(results[0].accountNumber == user.accountNumber){
                const toaccname = results[0].name;
                query1="update dashboard set balance=balance + " + user.amount +" where
                accountNumber=?";
                query2="update dashboard set balance=balance - " + user.amount +" where
                accountNumber=?";
                query3="insert into
                transaction(accountNumber,transtype,fromacc,toacc,amount,dateo,description)
                values(?,?,?,?,?,?,?)";
            }
        }
    })
});
```

```
query4="insert into
transaction(accountNumber,transtype,fromacc,toacc,amount,dateo,description)
values(?,debit',?,?,?,?,?);
connection.query(query1,[user.accountNumber],(err,results)=>{
    if(err) throw err;
connection.query(query2,[user.fromAccountNumber],(err,results)=>{
    if(err) throw err;

connection.query(query3,[user.accountNumber,user.fromname,toaccname,user.amoun
t,todayDate,user.description],(err,results)=>{
    if(err) throw err;

connection.query(query4,[user.fromAccountNumber,user.fromname,toaccname,user.a
mount,todayDate,user.description],(err,results)=>{
    if(err) throw err;
        });
    });
});
});
return res.status(200).json({message:"sucessful"})
else{
    return res.status(400).json({message:"something went wrong"});
}
}
});
});
});
```

10. Post API ‘Forgot Password’

```
router.post('/forgotPassword',(req,res)=>{
    const user=req.body;
    query= "select email,password from customer where email=?";
    connection.query(query,[user.email],(err,results)=>{
        if(!err){
            if(results.length <=0){
                return res.status(400).json({message:"Email Does not Exist"});
            }
            else{
                var mailOptions={
                    from: process.env.EMAIL,
                    to: results[0].email,
                    subject: "Password by-SPOTLIGHT",
                    html:"<p><b>Your Login Details for
SPOTLIGHT</b><br><b>EMAIL:</b>" +results[0].email+"<br><b>PASSWORD</
b>" +results[0].password+"</p>"
                };
                transporter.sendMail(mailOptions,function(err,info){
                    if(err){
                        console.log("Error");

```

```
        }
    else{
        console.log("email sent" + info.response);
    }
});
return res.status(200).json({message:"Password sent successfully to your
email"});
}
}
else{
    return res.status(500).json(err);
}
});
});
});
```

11. Post API ‘update’

```
router.post('/update',(req,res)=>{
let user = req.body;
var query= "update customer set "+user.field+"=? where accountNumber=?";
connection.query(query,[user.tovalue,user.accountNumber],(err,results)=>{
if(!err){
    if(results.affectedRows == 0){
        return res.status(404).json({message:"user id does not exist"});
    }
    return res.status(200).json({message: " user updated succesfully"});
}
else{
    console.log(err);
    return res.status(500).json(err);
}
});
});
```

12. Post API ‘change Password’

```
router.post('/changePassword',auth.authenticateToken,(req,res)=>{
const user = req.body;
const email = res.locals.email;
var query="select accountNumber,password,email,status from customer where
email=? and password=?";
console.log("1234");
connection.query(query,[email,user.oldPassword],(err,results)=>{
if(!err){
    if(results.length<=0){
        return res.status(400).json({message:"Incorrect old Password"});
    }
    else if(results[0].Password == user.oldPassword){
        query = "update customer set password=? where email=?";
```

```
connection.query(query, [user.newPassword,email],(err,results)=>{
  if(!err){
    return res.status(200).json({message: "Password updated successfully"});
  }
  else{
    return res.status(500).json(err);
    console.log("here");
  }
})
}
else{
  return res.status(400).json({message:"something went wrong"})
}
}
else{
  return res.status(500).json(err);
  console.log("there");
}
});
});
```

CHAPTER 6

INTERPRETATION OF RESULTS

Spotlight Bank Management system uses HTML, CSS and JavaScript for the fronted and , Node.js and MYSQL for the backend

- 1. User interface:** The use of HTML, JavaScript, and CSS for the front-end results in a user-friendly and visually appealing interface for the bank's customers and employees. The interface includes features such as forms for creating new accounts or updating customer information, tables or graphs for displaying transaction data, and buttons or links for navigating between pages.
- 2. Customer data storage:** The customer table in the database is used to store important information about the bank's customers, including their names, addresses, contact information, and account numbers. This information is likely used to help bank employees manage customer accounts and provide customer service.
- 3. Card information storage:** The cards table in the database stores information about the customer's card, such as the card number, expiration date, and card type. This information is important for managing card transactions and ensuring that customers are able to use their cards without issue.
- 4. Transaction data storage:** The transaction table in the database stores all the transaction data that takes place in the bank, including deposits, withdrawals, and transfers. This information is important for tracking account balances, identifying potential fraud or errors, and providing customers with transaction histories.
- 5. Dashboard:** The dashboard table in the database likely stores account numbers and balances, which can be used to quickly view and manage account information. This information may be used to monitor account activity, identify potential issues or opportunities, and help bank employees make informed decisions.

Overall, the use of HTML, JavaScript, and CSS for the front-end and Node.js and SQL for the back-end likely results in a robust and efficient bank management system that provides users with a user-friendly interface and access to important data about customers, cards, transactions, and accounts.

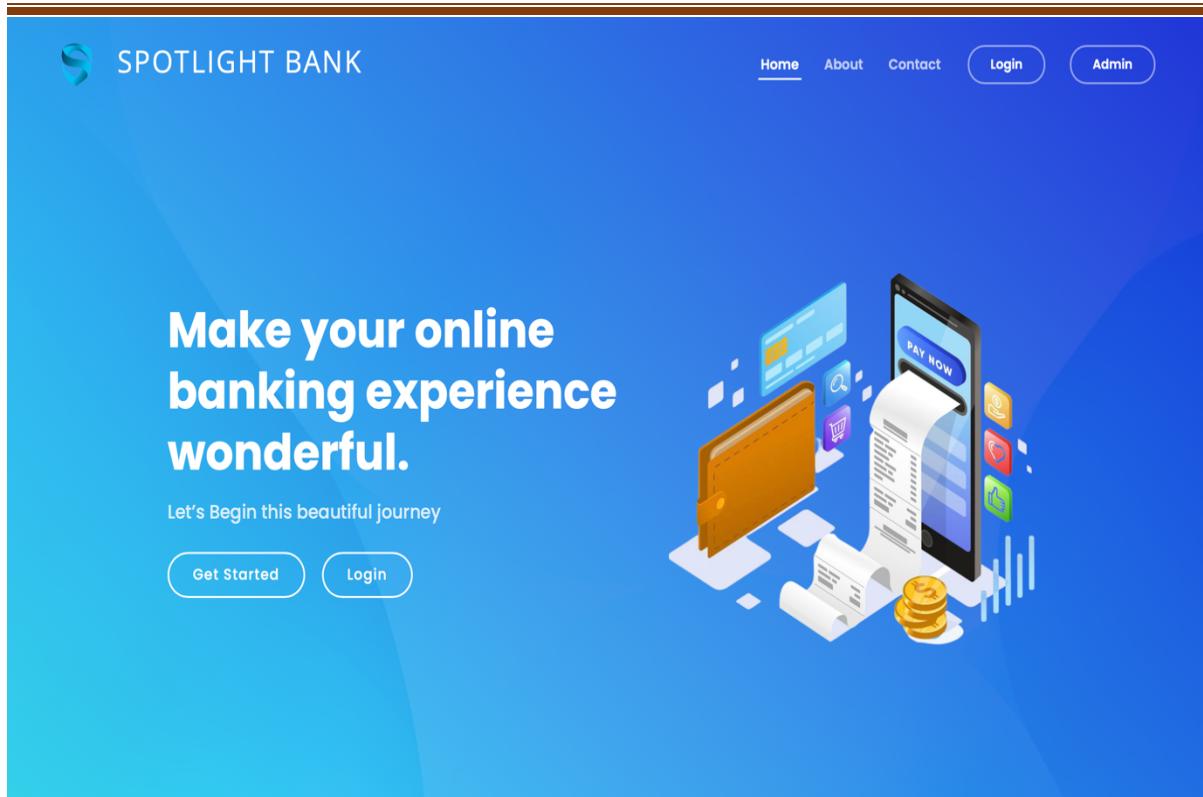


Figure 6.1 Main Window

The 'About Us' window has a dark blue header bar with the 'SPOTLIGHT BANK' logo and navigation links for 'Home', 'About' (underlined), 'Contact', and 'Login/Admin'. The main content area has a light blue background. At the top, the title 'ABOUT US' is centered. To the left, there is a paragraph of text: 'Welcome to Spotlight Bank! We are a trusted financial institution committed to providing reliable and secure banking services to our customers. Our team of experienced professionals is dedicated to delivering exceptional customer service and supporting you every step of the way. Whether you need assistance with opening an account, managing your finances, or navigating our online banking platform, we are here to help. We take the security and privacy of your information seriously. We employ the latest technology and security measures to protect your data and ensure that your online transactions are safe and secure.' Below this text is another paragraph: 'Thank you for choosing us as your banking partner. We look forward to serving you and helping you achieve your financial goals.' To the right of the text is a central graphic showing two people interacting with a large smartphone. The phone screen displays the word 'ABOUT US' and several icons related to banking and finance. Below the graphic are three categories with corresponding icons: 'Money Savings' (wallet icon), 'Online Shoppings' (shopping cart icon), and 'Credit / Debit Cards' (credit card icon). A small upward-pointing arrow is located in the bottom right corner of the page.

Figure 6.2 About Us Window

SPOTLIGHT BANK MANAGEMENT SYSTEM

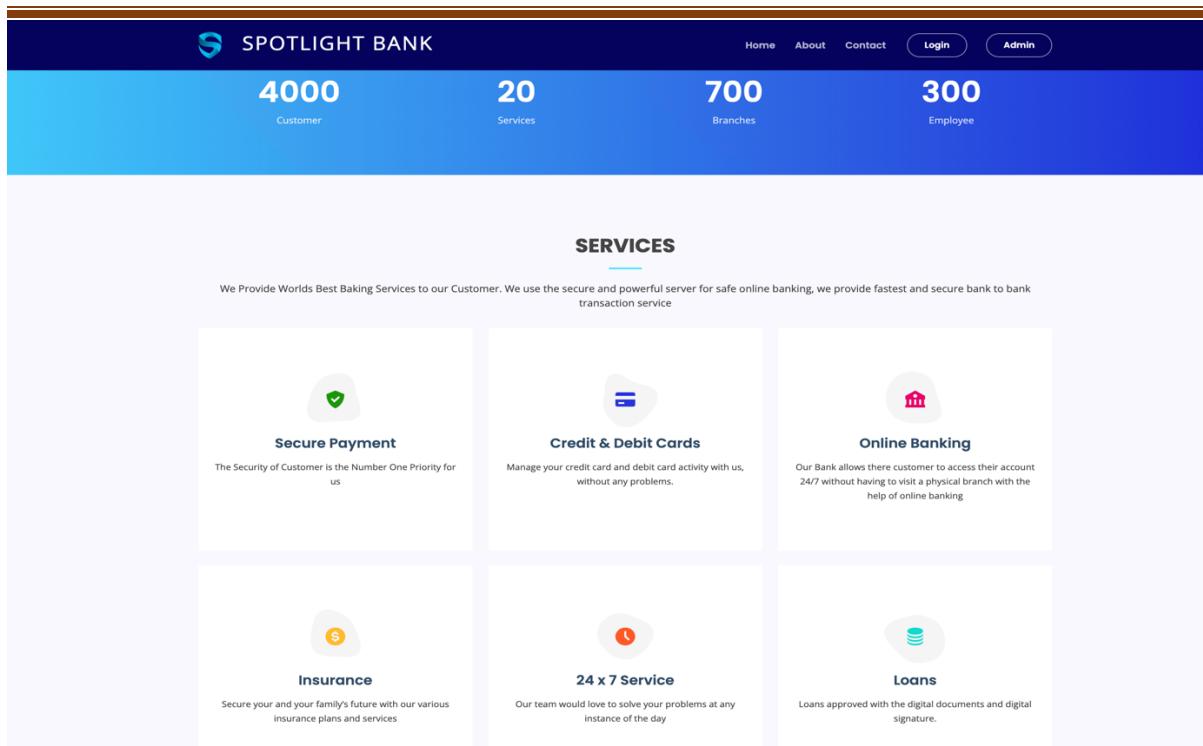
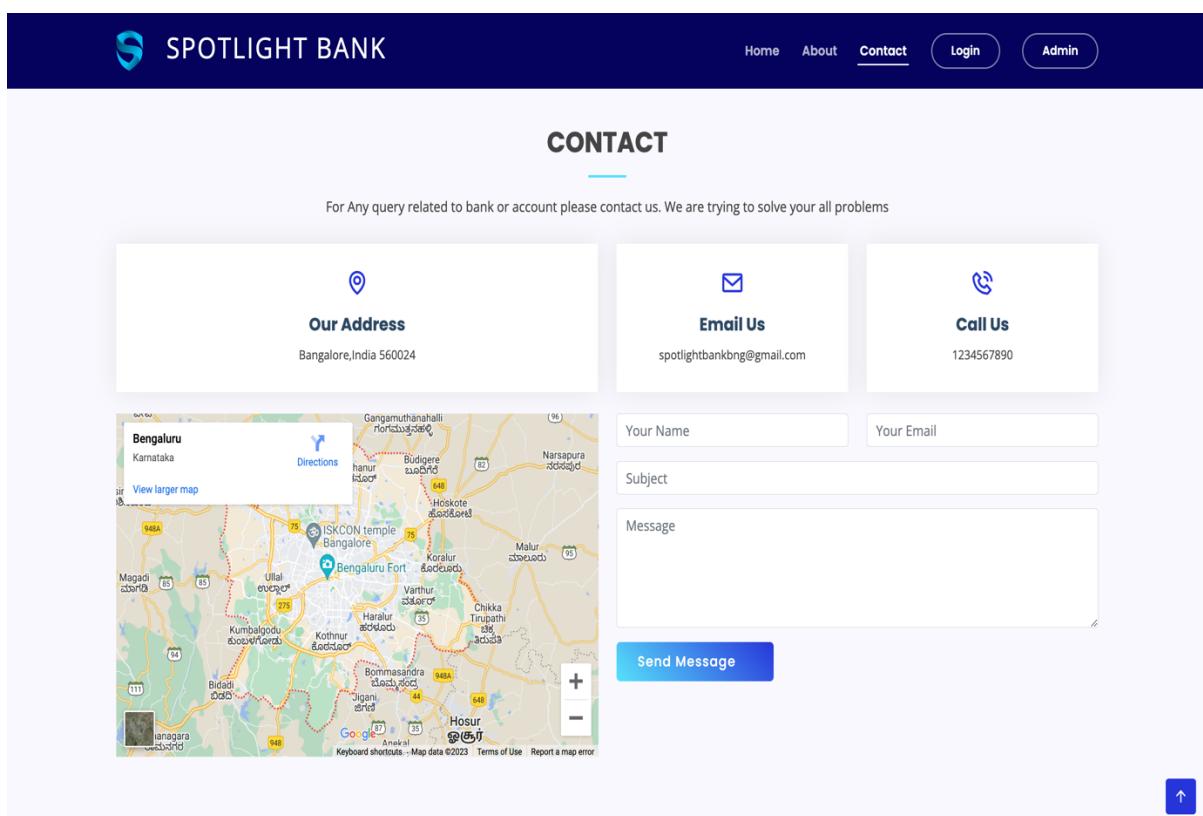


Figure 6.3 Services Window



The Contact Us window provides multiple ways to get in touch with the bank. It includes:

- Our Address**: Located in Bangalore, India 560024.
- Email Us**: spotlightsbankbng@gmail.com
- Call Us**: 1234567890
- A **Map** showing the location of Bangalore, including landmarks like ISKCON temple and Bangalore Fort.
- An **Contact Form** with fields for Name, Email, Subject, Message, and a Send Message button.

Figure 6.4 Contact Us Window

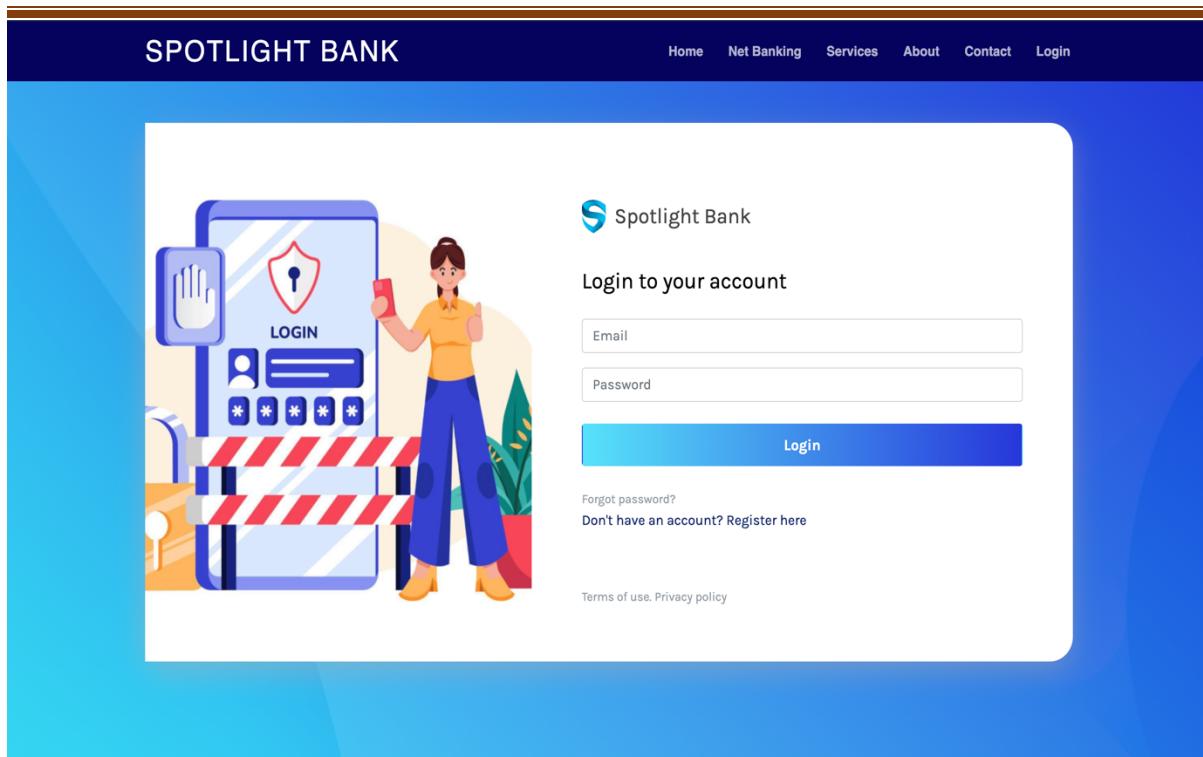


Figure 6.5 Login Page

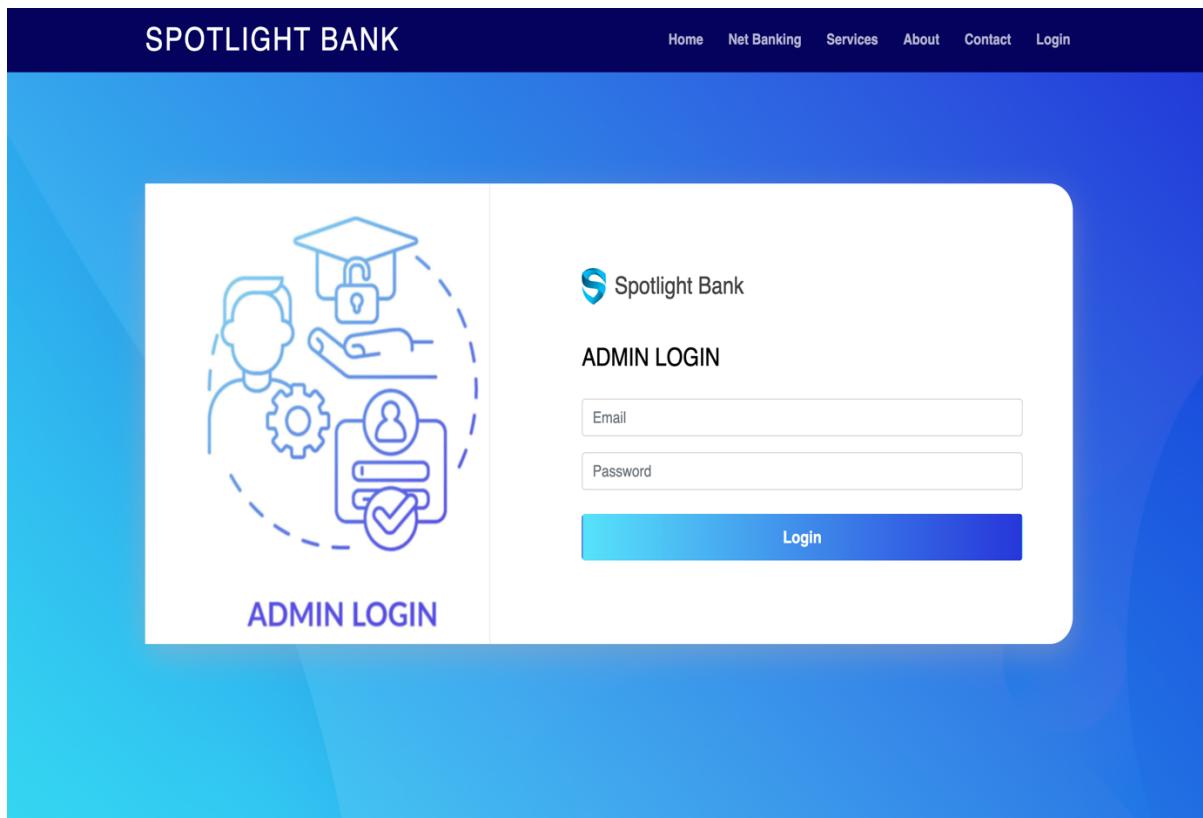


Figure 6.6 Admin Login Page

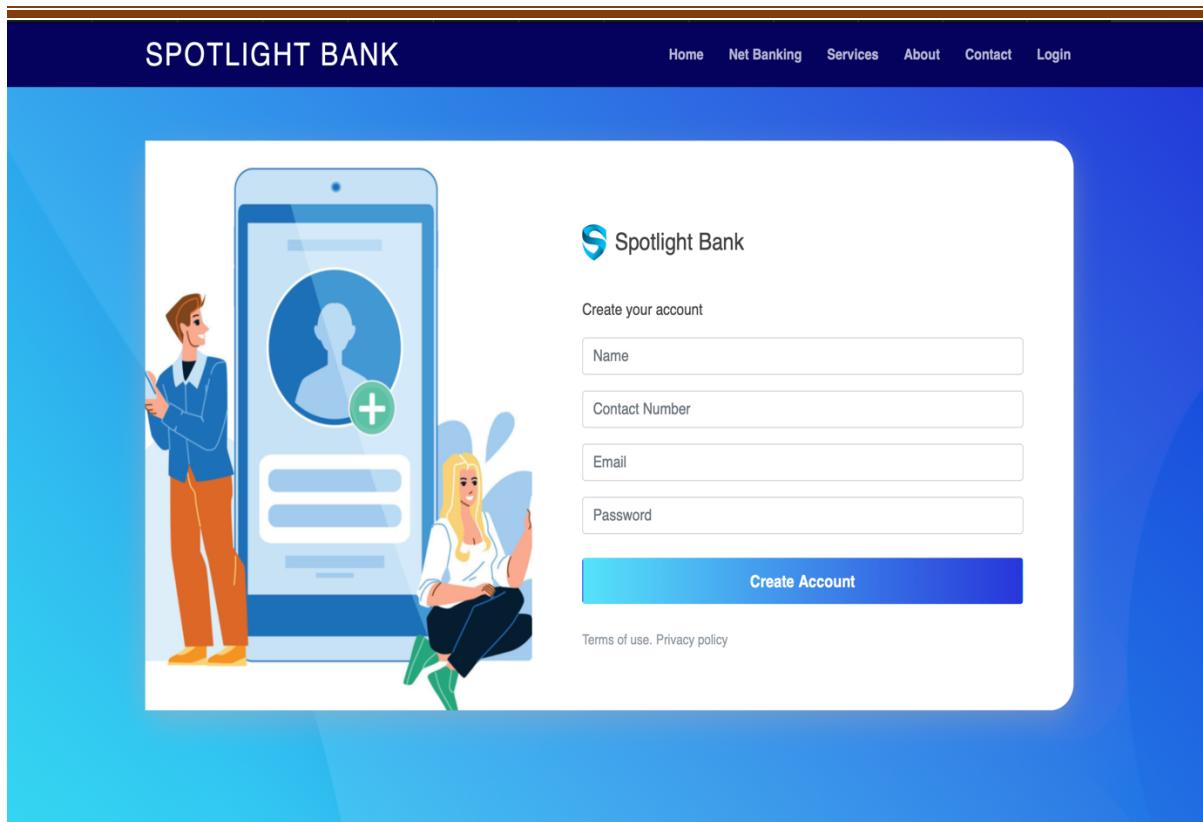


Figure 6.7 Register User Page

Transaction Id	From Account No	To Account No	Date	Amount	Description
9	Shreya BMC	Shreyas	2023-03-08	5000	scam money
2	Shreyas	Admin	2023-03-07	0	initial
1	Admin	Shreyas	2023-03-07	0	initial

Figure 6.8 Customer Dashboard Page

[☰ Spotlight Bank](#)

Transfer Money

 Enter Account No... ₹ Enter Amount... Description...**Pay Money**

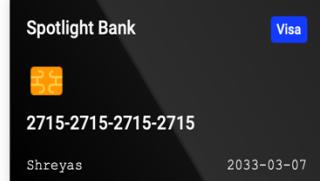
Spotlight Bank ©

[Privacy](#) [Terms](#)**Figure 6.9** Customer Transaction Page[☰ Spotlight Bank](#)

Spotlight Bank

Cards

Your Debit Card



Your Debit Card Details

Account Number	:	7048704870
Debit Card Number	:	2715-2715-2715-2715
Name	:	Shreyas
CVV Number	:	980
Issued Date	:	2023-03-07
Expiry Date	:	2033-03-07
Card Status	:	active

Spotlight Bank ©

[Privacy](#) [Terms](#)**Figure 6.10** Customer Card Info Page

SPOTLIGHT BANK MANAGEMENT SYSTEM

accountNumber	name	contactNumber	email	password	status	role
1	Admin	123456789	admin@gmail.com	admin	true	admin
2	shreyas	123456789	shreyas@gmail.com	shreyas	true	user
1234567890	shreyas	1234567890	asfsdgl@gmail.com	test	true	user
1471147114	shubh	1234567890	shubh12345@gmail.com	shubh	true	user
5601225544	Shreya BMC	1234567890	shreyas.naidu2003@gmail.com	bmc	true	user
6908622524	shal	7076597	shali@gmail.com	Shal@6	true	user
7048704870	Shreyas	1234567890	rshreyas2003@gmail.com	shreyas	true	user
8766876687	shreyas	1234567890	asfsdkjgl@gmail.com	test	true	user

Figure 6.11 Customer Table Database

accountNumber	cardName	cardNumber	cvv	issuedDate	expiryDate	cardStatus
7048704870	Shreyas	2715271527152715	980	2023-03-08	2033-03-08	active
1471147114	shubh	5690569056905690	861	2023-03-08	2033-03-08	active
5601225544	Shreya BMC	6973952799598277	156	2023-03-09	2033-03-09	active
6908622524	shal	8134388865396275	168	2023-03-08	2033-03-08	active

Figure 6.12 Cards Table Database

transactionId	accountNumber	transtype	fromacc	toacc	amount	dateo	description
1	7048704870	credit	Admin	Shreyas	0	2023-03-08	initial
2	7048704870	debit	Shreyas	Admin	0	2023-03-08	initial
3	1471147114	credit	Admin	shubh	0	2023-03-08	initial
4	1471147114	debit	shubh	Admin	0	2023-03-08	initial
5	6908622524	credit	Admin	shal	0	2023-03-08	initial
6	6908622524	debit	shal	Admin	0	2023-03-08	initial
7	5601225544	credit	Admin	Shreya BMC	0	2023-03-09	initial
8	5601225544	debit	Shreya BMC	Admin	0	2023-03-09	initial
9	7048704870	credit	Shreya BMC	Shreyas	5000	2023-03-09	scam money
10	5601225544	debit	Shreya BMC	Shreyas	5000	2023-03-09	scam money

Figure 6.13 Transaction Table Database

accountNumber	balance
7048704870	15000
1471147114	10000
6908622524	10000
5601225544	5000

Figure 6.14 Dashboard Table Database

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in a bank. It keeps the day by day tally record as a complete batting. It can keep the information of Account type, account opening form, Deposit, Withdrawal, and Searching the transaction, Transaction report. Individual account opening form. Group Account. The exciting part of this project is; it displays Transaction reports, Statistical Summary of Account type and Interest Information.

Online banking is an innovative tool that is fast becoming a necessity. It is a successful strategic weapon for banks to remain profitable in a volatile and competitive marketplace of today. If proper training should be given to customer by the bank employs to open an account will be beneficial secondly the website should be made friendlier from where the first time customers can directly make and access their accounts.

In the future version of the project, we can add additional layers of security like hashing and salting before storing the data in the database. Additional features like KYC can also be added to improve the authenticity of the Spotlight Bank Management System.

REFERENCES

1. <https://www.w3schools.com>
2. <https://getbootstrap.com>
3. <https://www.wix.com>
4. <https://github.com>
5. [https://www.academia.edu/9349358/54024405 project report banking management system 130507125726 phpapp02](https://www.academia.edu/9349358/54024405_project_report_banking_management_system_130507125726_phpapp02)
6. https://www.slideshare.net/IbadullahKhan6/bank-management-system-desktop-application?from_search=9
7. <https://www.scribd.com/document/89533666/Report-on-Banking-Management-System#>
8. <https://www.youtube.com>