VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

An Internship/Professional work Report

on

Banking Management System

Submitted in partial fulfillment of the requirements for the final year degree in
Bachelor of Engineering in Computer Science and Engineering
of Visvesvaraya Technological University, Belagavi

Submitted by

Sachin Kumar 1RN20CS119

Carried out at

TechieAid

Under the Guidance of:

Internal Guide:
Prof.Mamatha Jajur

Assistant Professor
Dept. of CSE, RNSIT

External Guide:
Kiran Gopalakrishna

Company
Bengaluru

Department of Computer Science and Engineering
RNS Institute of Technology
(Accredited by NBA upto 30-06-2025)
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

2023-2024

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(Accredited by NBA upto 30-06-2025)

CERTIFICATE

Certified that the Internship/Professional Practice work entitled **"Banking Management System"** has been successfully carried out at **company name** by **Sachin Kumar** bearing USN **1RN20CS119** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements of final year degree in **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2023-2024**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Prof.Mamatha Jajur	**Dr. Kiran P**	**Dr. Ramesh Babu H S**
Assistant Professor	Professor and HOD	Principal
Dept. of CSE	Dept. of CSE	RNSIT
RNSIT	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1.	
2.	

Figure 1: Internship certificate

TechieAid

Date : 21st November 2023

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Mr. Sachin Kumar** a student from RNSIT with USN : 1RN20CS119 has successfully completed Internship program in **Advanced Full Stack Web Development** from August 2023 till September 2023.

During this period of Internship, Mr. Sachin Kumar worked on various areas of software development including project work titled **"Banking Management System"**. He has successfully met the objectives of the internship program and his conduct was found to be satisfactory.

We wish him all the best for future endeavors.

Kiran Gopalakrishna,
Founder & CEO,
TechieAid, Bangalore

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to Management and **Dr. Ramesh Babu H S**, Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P**, Professor & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Prof. Mamatha Jajur** , Assistant Professor, Department of CSE, RNSIT, Bangalore, for her able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of Department of Computer Science & Engineering, RNSIT, Bengaluru-98 for their constant support and encouragement.

Abstract

A bank management system is a software application that is used by banks to manage their daily operations, such as account management, financial transactions, and customer information management. The system typically includes modules for customer account management, online banking, loan management, and reporting. The system can also include advanced features such as fraud detection, risk management, and compliance with regulatory requirements. The goal of a bank management system is to automate and streamline the bank's operations, while also providing accurate and timely information to management and customers.

A bank management system typically includes several key modules that are used to manage different aspects of the bank's operations. Some of the key modules include:

Account Management: This module is used to manage customer accounts, including deposit and loan accounts. It allows bank staff to view account balances, transaction history, and other account information.

Online Banking: This module allows customers to access their accounts through the internet, perform transactions, and view account information.

Loan Management: This module is used to manage the bank's loan portfolio, including loan origination, underwriting, and servicing.

Reporting: This module generates various reports that are used by bank management to track the performance of the bank.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Database Management System

1.1 Introduction

Databases and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what a database is. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database. The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database

- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

1.2 History of DBMS

The history of database management systems (DBMS) can be traced back to the 1960s, when IBM developed the Integrated Data Store (IDS) for the U.S. Air Force. This was one of the first database management systems to be developed and used in a production environment.

In the 1970s, the relational model for databases was introduced by Dr. E.F. Codd at IBM. He proposed a new way of organizing data using tables with rows and columns, and a structured query language (SQL) for data manipulation. This model became the basis for the development of relational database management systems (RDBMS), such as IBM's System R and later, Oracle.

In the 1980s, the popularity of RDBMSs grew as personal computers became more powerful and affordable. This led to the development of several new RDBMSs, such as SQL Server, MySQL, and PostgreSQL.

In the 1990s, the internet and the World Wide Web emerged as a major new platform for information management and communication. This led to the development of new types of DBMSs, such as document databases and key-value stores, which are better suited for storing and retrieving unstructured data.

In recent years, NoSQL databases have gained popularity for their ability to handle large amount of unstructured data and also for their scalability, performance and ease of handling distributed data.

Overall, the history of DBMSs has been one of constant evolution, driven by advances in technology, changes in the way data is used and stored, and the emergence of new platforms and

use cases.

1.3 Applications of DBMS

Applications where we use Database Management Systems are:

- **Telecom:** There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry:** Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System:** For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
- **Education Sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online Shopping:** You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

Chapter 2

Tools and Technologies Used

2.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as ¡img /¿ and ¡input /¿ introduce content into the page directly. Others such as ¡p¿...¡/p¿ surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

2.2 Bootstrap-A CSS Framework

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other

media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Bootstrap is a free and open-source front-end web framework used for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 111,600 stars and 51,500 forks.

2.3 Javascript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and this is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

2.4 PHP

PHP (Hypertext Preprocessor) is a server-side scripting language that is commonly used to create dynamic web pages. It is open-source software, which means that it can be freely modified and

distributed. PHP is often used in conjunction with a web server like Apache and a database management system like MySQL, to create dynamic web applications.

PHP code can be embedded within HTML code, making it easy to create interactive and dynamic web pages. Some of the features of PHP include: Variables: PHP supports a wide range of data types, including integers, strings, and arrays. Functions: PHP has a large number of built-in functions that can be used to perform a wide range of tasks, such as working with strings, arrays, and dates. Control Structures: PHP supports all of the standard control structures, such as if/else statements and loops. Error handling: PHP has built-in error handling capabilities, which can be used to handle errors and exceptions in a consistent and controlled manner. Database connectivity: PHP has built-in support for connecting to a wide range of databases, such as MySQL, Oracle, and Microsoft SQL Server. Security: PHP provides several built-in functions and extensions to help secure your code from common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

2.5 MySQL

MySQL is a widely used, open-source relational database management system (RDBMS). It is designed to store and manage large amounts of data in an efficient and organized manner. MySQL is particularly popular for web-based applications, due to its speed, reliability, and ease of use. Some of the features of MySQL include: SQL support: MySQL uses the SQL (Structured Query Language) to interact with the database, allowing developers to easily create, read, update, and delete data. Tables and relationships: MySQL allows you to create multiple tables, each with its own structure and data, and to define relationships between them. Indexes: MySQL allows you to create indexes on specific columns in a table to improve query performance. Stored procedures and triggers: MySQL allows you to create stored procedures and triggers to automate repetitive tasks and enforce data integrity. Security: MySQL supports a variety of security features, such as user management, encryption, and access control, to help protect data from unauthorized access.

Chapter 3

Resource Requirements

3.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.Table 3.1 gives details of hardware requirements.

Table 3.1: Hardware Requirements

Processor	Intel Core i3 processor
Processor Speed	1.70 GHz
RAM	4 GB
Storage Space	40 MB
Monitor Resolution	1024*768 or 1336*768 or 1280*1024

3.2 Software Requirements

The software requirements are description of features and functionalities of the system.Table 3.2 gives details of software requirements.

Table 3.2: Software Requirements

Operating System	Windows 7 Above
IDE	Microsoft Visual Studio with C++ 2022
Other Requiremnets	Xampp,MySQLWorkbench,NotePad++.

3.3 Functional Requirements

3.3.1 Major Entities

branch: This includes information about the different branches of the bank present .

Feedback:This includes information about the feedback presented by the customers to the bank.

Login: This includes information about the Login details of all the customers who have their account in the bank.

Noties: This includes information about the all the notification sent by the bank to the customers regarding their account.

otheraccounts: This includes information about the account details of the customers who have account in different bank.

Transaction: This includes information about the all the transaction details of the customers .

Useraccounts:This includes the information about the details of the customers who have their account in the bank.

3.3.2 End User Requirements

1. user requirements for a bank management system will vary depending on the specific needs of the bank and its customers. However, some common requirements for such a system include:
2. User-friendly interface: The system should be easy to use and navigate for both bank staff and customers.
3. Data security: The system should be designed to protect sensitive customer information and ensure compliance with data privacy regulations.
4. Scalability: The system should be able to handle large volumes of data and transactions, and be able to scale as the bank's operations grow.
5. Integration with other systems: The system should be able to integrate with other systems used by the bank, such as core banking systems, accounting systems, and customer relationship management (CRM) systems.
6. Reporting and analytics: The system should provide a variety of reports and analytics tools to help bank management track the performance of the bank and make informed decisions.
7. Mobile access: The system should allow customers to access their accounts and perform transactions through their mobile devices.

Chapter 4

Database Design

4.1 Entities, Attributes and Relationships

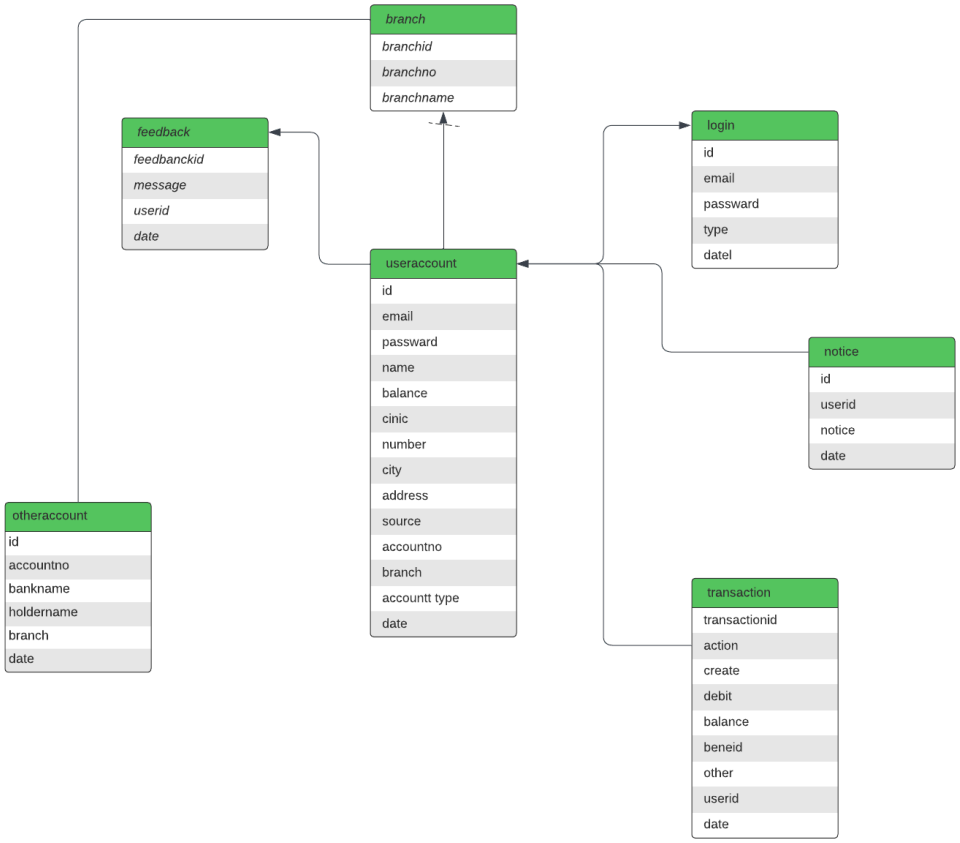
The database, called data, will have seven tables, branch, feedback, login, notice, otheraccounts, transaction, useraccount. Each will hold information about banking system . The two tables will be linked through a foreign key. The table has the following fields:

4.2 Identify Major entities, attributes and relationships

- Login page to give access to priviledged Admin.
- Adding car details details by the Admin.
- Changing the id and password of the staffs. which stores the details of transaction.
- Admin can check all details and modify them.
- Admin can delete the entities in the tables.
- Easy search facilities to get the reqiured information.

4.3 ER Schema

Figure 4.1: Entity Relationship Diagram



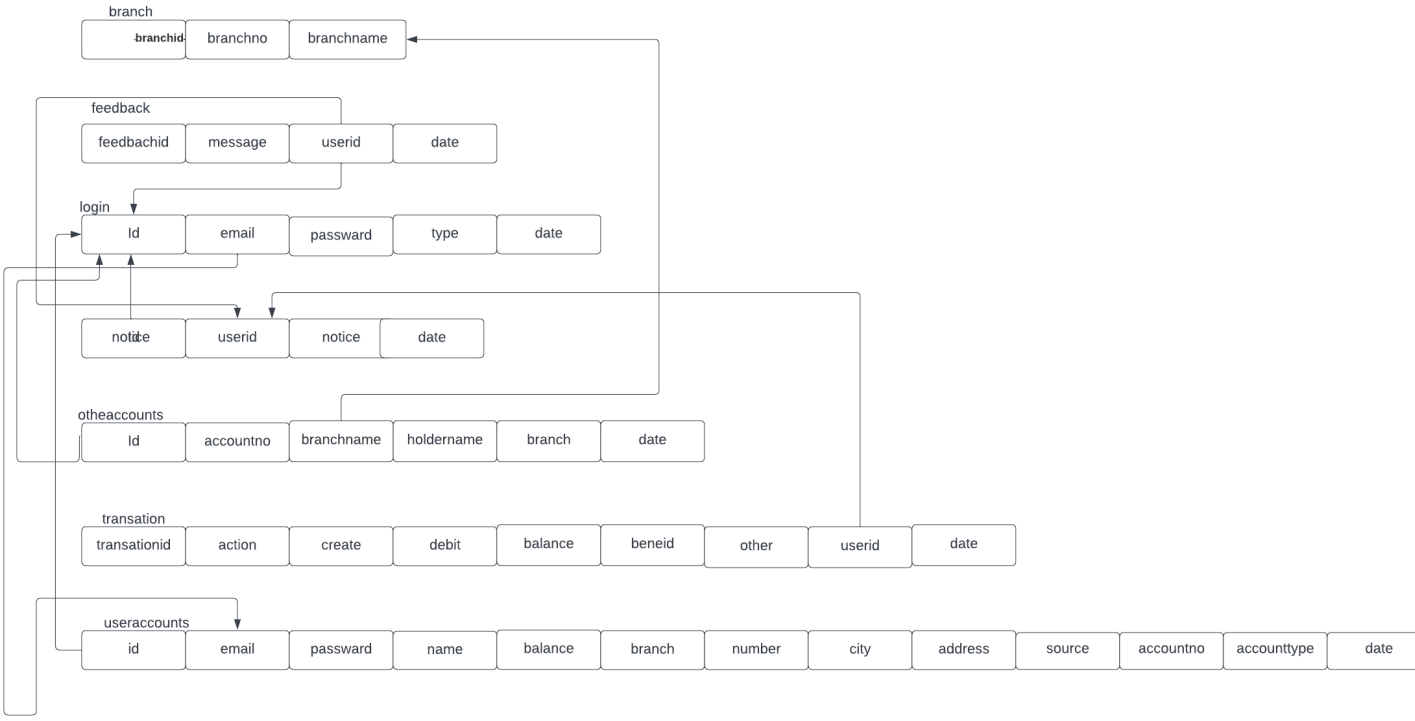
This above figure 4.2 shows ER schema diagram of our banking management system. It depicts relation

between entity like branch, feedback, login, notice, otheraccounts, transaction, useraccount. etc.Booking has a 'have' relation banking mangement system

Similarly bankind related is with transmission type,engine type with 'has' type of relation.

4.4 Schema Diagram

Figure 4.2: Relational Schema



- Above figure schema shows relational schema of our project
- Contain 7 schemas each having foreign keys and primary keys
- banking mangement system schema has id as a primary key,similary other tables have their own primary key which is always unique and not null.

Chapter 5

Implementation

5.1 PHP

```
<?php
 session_start  () ;
if (! isset ($_SESSION['userId'])){  header(' location : login .php') ;}
?>
<!DOCTYPE html>
<html>
<head>
   < title >Banking</title>
   <?php require  ' assets / autoloader .php';  ?>
   <?php require  ' assets /db.php';  ?>
   <?php require  ' assets / function .php';  ?>

</head>
<body  style="background-size: 100%" class="bg-gradient-seconday">
<nav  class ="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
 <a  class ="navbar-brand" href="#">
      <img src="images/logo.png"  style ="object- fit : cover; object - position : center   center"
     width="30" height="30"  class ="d-inline-block  align -top"  alt ="">
    <!--  <i  class ="d-inline-block   fa  fa-building  fa-fw"></i> --><?php echo bankname;
     ?>
   </a>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#
  navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
   aria-label="Toggle navigation">
   <span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
   <ul class="navbar-nav mr-auto">
      <li class="nav-item ">
         <a class="nav-link active" href="index.php">Home <span class="sr-only">(current)
   </span></a>
      </li>
      <li class="nav-item "> <a class="nav-link" href="accounts.php">Accounts</a></li>
      <li class="nav-item "> <a class="nav-link" href="statements.php">Account
   Statements</a></li>
      <li class="nav-item "> <a class="nav-link" href="transfer.php">Funds Transfer</a
   ></li>
      <!-- <li class="nav-item "> <a class="nav-link" href="profile.php">Profile</a></li
   > -->



   </ul>
   <?php include 'sideButton.php'; ?>


</div>
</nav><br><br><br>
<div class="row w-100" >
   <div class="col" style="padding: 22px;padding-top: 0">
      <div class="jumbotron shadowBlack" style="padding: 25px;min-height: 241px;max-height
      : 241px">
   <h4 class="display-5">Welecome to XYZ Banko</h4>
   <p class="lead alert alert-warning"><b>Latest Notification:</b>
```

```
<?php
    $array = $con->query("select * from notice where userId = '$_SESSION[userId]' order
  by date desc");
    if ($array->num_rows > 0)
    {
      $row = $array->fetch_assoc();
      // {
        echo $row['notice'];
      // }
    }
    else
      echo "<div class='alert alert-info'>Notice box empty</div>";
  ?></p>

</div>
        <div id="carouselExampleIndicators" class="carousel slide my-2 rounded-1
  shadowBlack" data-ride="carousel">
          <div class="carousel-inner">
          <div class="carousel-item active">
            <img class="d-block w-100" src="images/bank1.jpg" alt="First slide" style="
  max-height: 250px">
          </div>
          <div class="carousel-item">
            <img class="d-block w-100" src="images/bank2.jpg" alt="Second slide" style="
  max-height: 250px">
          </div>
          <div class="carousel-item">
            <img class="d-block w-100" src="images/bank3.png" alt="Third slide" style="
  max-height: 250px">
          </div>

        </div>
        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"
```

```
    data-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button"
    data-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
      </div>
  </div>
<div class="col">
    <div class="row" style="padding: 22px;padding-top: 0">
      <div class="col">
        <div class="card shadowBlack ">
          <img class="card-img-top" src="images/journal.jpg" style="max-height: 155px;
    min-height: 155px" alt="Card image cap">
          <div class="card-body">
            <a href="accounts.php" class="btn btn-outline-info shadow btn-block">Account
     Summary</a>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="card shadowBlack ">
          <img class="card-img-top" src="images/mtransfer.jpg" alt="Card image cap" style
    ="max-height: 155px;min-height: 155px">
        <div class="card-body">
          <a href="transfer.php" class="btn btn-outline-info shadow btn-block">Transfer
    Money</a>
        </div>
        </div>
      </div>
```

```
    </div>
    <div class="row" style="padding: 22px">
      <div class="col">
        <div class="card shadowBlack ">
          <img class="card-img-top" src="images/notif.gif" style="max-height: 155px;min-
    height: 155px" alt="Card image cap">
          <div class="card-body">
            <a href="notice.php" class="btn btn-outline-primary btn-block">Check
    Notification</a>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="card shadowBlack ">
          <img class="card-img-top" src="images/send_mail.gif" alt="Card image cap" style
    ="max-height: 155px;min-height: 155px">
          <div class="card-body">
            <a href="feedback.php" class="btn btn-outline-primary btn-block">Contact Us</a
    >
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

5.2 Create Table Commands

```
CREATE TABLE ‘branch‘ (
  ‘branchId‘ int(11) NOT NULL,
  ‘branchNo‘ varchar(111) NOT NULL,
  ‘branchName‘ varchar(111) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
INSERT INTO ‘branch‘ (‘branchId‘, ‘branchNo‘, ‘branchName‘) VALUES
(1, ’100101’, ’Dera Ghazi Khan’),
(2, ’100102’, ’Multan’);


CREATE TABLE ‘feedback‘ (
  ‘feedbackId‘ int(11) NOT NULL,
  ‘message‘ text NOT NULL,
  ‘userId‘ double NOT NULL,
  ‘date‘ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
INSERT INTO ‘feedback‘ (‘feedbackId‘, ‘message‘, ‘userId‘, ‘date‘) VALUES
(1, ’This is testing message to admin or manager by fk’, 1, ’2017-12-15 04:30:48’),
(3, ’This is testing message to admin or manager by fk’, 2, ’2017-12-15 04:30:48’),
(4, ’this is help card for admin’, 1, ’2017-12-17 06:45:20’);

CREATE TABLE ‘login‘ (
  ‘id‘ int(11) NOT NULL,
  ‘email‘ varchar(111) NOT NULL,
  ‘password‘ varchar(111) NOT NULL,
  ‘type‘ varchar(111) NOT NULL,
  ‘date‘ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
INSERT INTO ‘login‘ (‘id‘, ‘email‘, ‘password‘, ‘type‘, ‘date‘) VALUES
(1, ’cashier@cashier.com’, ’cashier’, ’cashier’, ’2017-12-15 04:36:27’),
(2, ’manager@manager.com’, ’manager’, ’manager’, ’2017-12-15 04:36:27’),
(3, ’sadfas@gmail.com’, ’sdfas’, ’type’, ’2017-12-16 07:13:12’),
(4, ’fkgeo@gmail.com’, ’asdfsa’, ’type’, ’2017-12-16 07:13:18’),
```

```
(6, 'cashier2@cashier.com', 'cashier2', 'cashier', '2017-12-16 07:14:47');

CREATE TABLE `notice` (
  `id` int(11) NOT NULL,
  `userId` varchar(111) NOT NULL,
  `notice` text NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
INSERT INTO `notice` (`id`, `userId`, `notice`, `date`) VALUES
(1, '1', 'Dear Customer! <br> OUr privacy policy is changed for account information get
    new prospectus from your nearest branch', '2017-12-14 13:11:46'),
(6, '2', 'Dear Ali,<br>\r\nOur privacy policy has been changed please visit nearest <kbd
    > MCB </kbd> branch for new prospectus.', '2017-12-16 06:29:23');

CREATE TABLE `otheraccounts` (
  `id` int(11) NOT NULL,
  `accountNo` varchar(111) NOT NULL,
  `bankName` varchar(111) NOT NULL,
  `holderName` varchar(111) NOT NULL,
  `balance` varchar(111) NOT NULL,
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
INSERT INTO `otheraccounts` (`id`, `accountNo`, `bankName`, `holderName`, `balance`, `
    date`) VALUES
(1, '12001122', 'UBL', 'Yaqoob Quraishi', '40800', '2017-12-14 11:55:07'),
(2, '12001123', 'HBL', 'Yousaf Khan', '71000', '2017-12-14 11:55:07'),
(3, '12001124', 'HBL', 'Yousaf Khan', '71000', '2017-12-14 11:55:07');

CREATE TABLE `transaction` (
  `transactionId` int(11) NOT NULL,
  `action` varchar(111) NOT NULL,
  `credit` varchar(111) NOT NULL,
  `debit` varchar(111) NOT NULL,
```

```
    `balance`  varchar(111)  NOT NULL,
    `beneId`  varchar(111)  NOT NULL,
    `other`  varchar(111)  NOT NULL,
    `userId`  int(11)  NOT NULL,
    `date`  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `useraccounts` (
    `id`  int(11)  NOT NULL,
    `email`  text  NOT NULL,
    `password`  text  NOT NULL,
    `name` varchar(111)  NOT NULL,
    `balance`  varchar(111)  NOT NULL,
    `cnic`  varchar(111)  NOT NULL,
    `number` varchar(111)  NOT NULL,
    `city`  varchar(111)  NOT NULL,
    `address`  varchar(111)  NOT NULL,
    `source`  varchar(111)  NOT NULL,
    `accountNo` varchar(111)  NOT NULL,
    `branch`  varchar(111)  NOT NULL,
    `accountType`  varchar(111)  NOT NULL,
    `date`  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Chapter 6

Results & Snapshots

Figure 6.1: Admin Log In

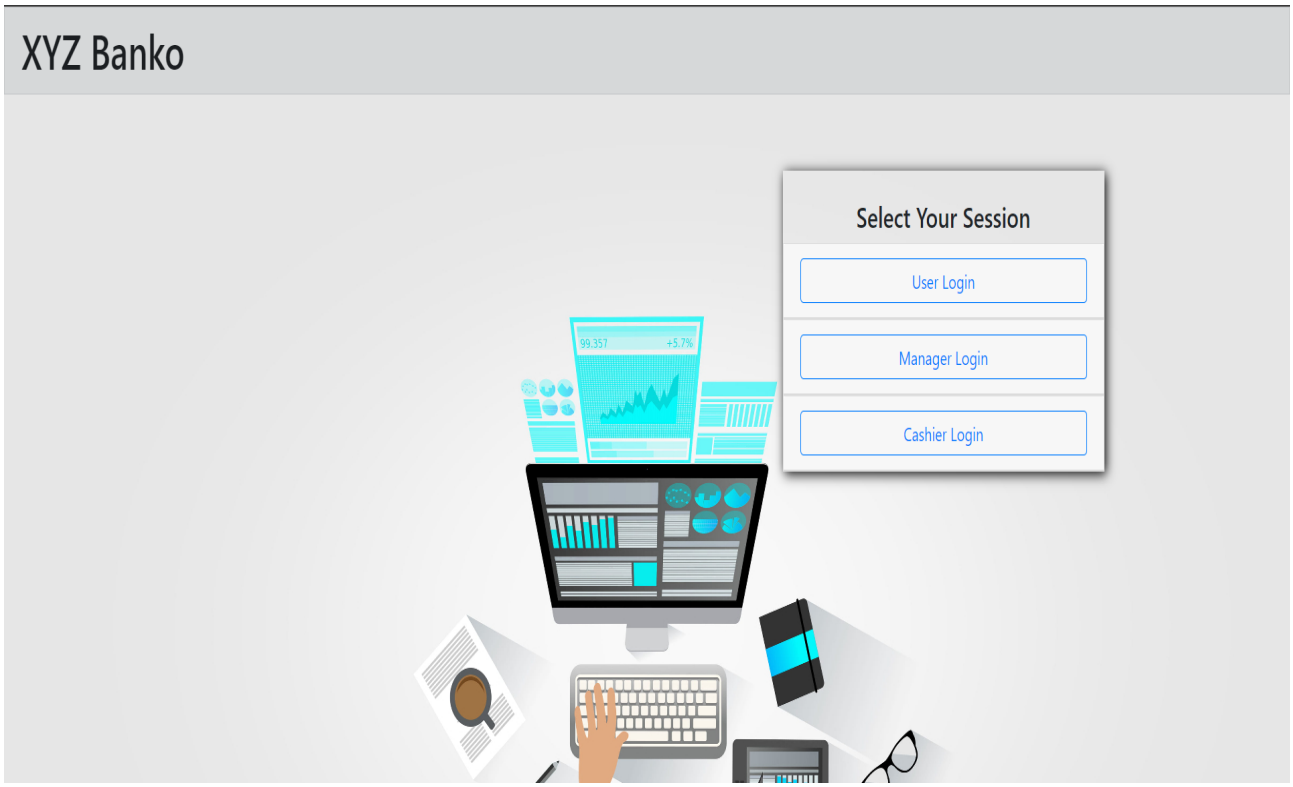


Figure 6.2: Landing Page

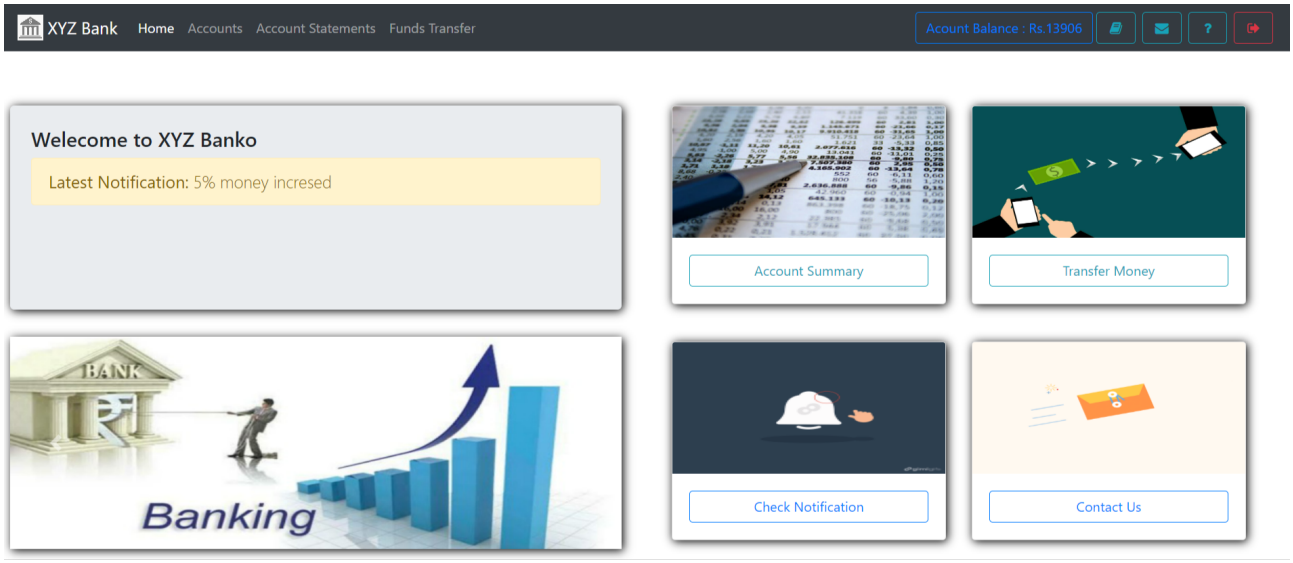


Figure 6.3: Admin Dashboard

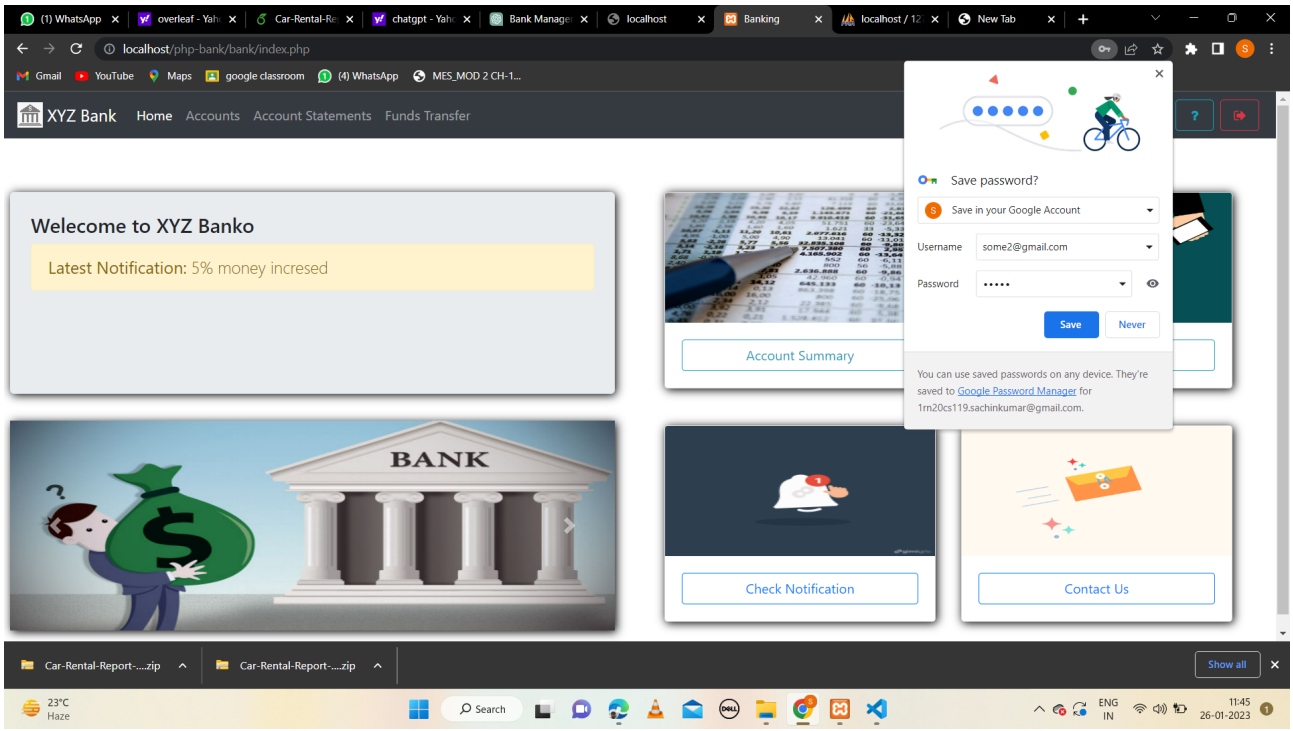


Figure 6.4: Pickup/dropoff

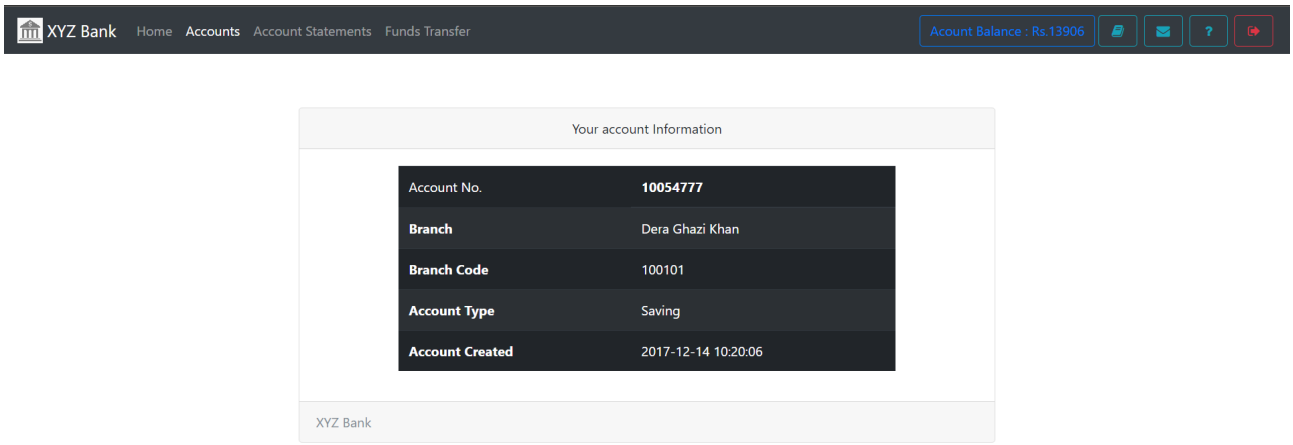
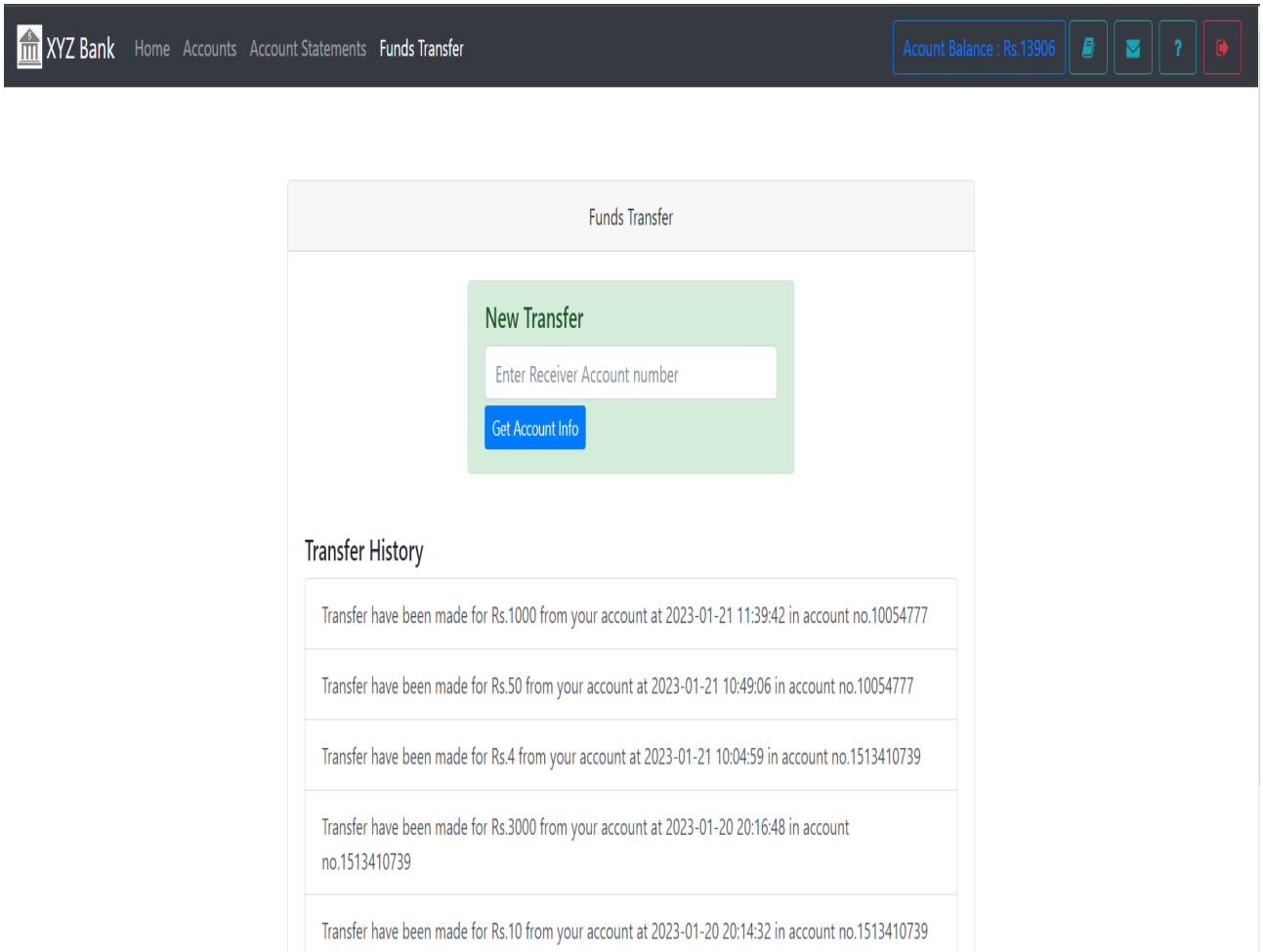


Figure 6.5: Cars



Chapter 7

Conclusion & Future Enhancements

7.1 Conclusion

In conclusion, a bank management system is a critical tool for banks to manage their daily operations and provide efficient services to their customers. It automates and streamlines various processes, from account management and financial transactions to compliance with regulatory requirements. The system provides a variety of features, including account management, online banking, loan management, reporting, fraud detection, risk management, and compliance. It also allows for integration with other systems used by the bank, such as core banking systems and CRM systems.

The system should be user-friendly, secure, and scalable, with a variety of reporting and analytics tools to help bank management make informed decisions. It should also comply with all relevant regulatory requirements, such as Anti-Money Laundering (AML) and Know Your Customer (KYC) regulations. With the right bank management system in place, banks can improve their efficiency, reduce errors, and provide better services to their customers.

7.2 Future Enhancements

There are several areas where future enhancements to bank management systems could be made to improve the efficiency and effectiveness of banking operations:

Artificial Intelligence (AI) and Machine Learning (ML): These technologies can be used to analyze customer data and detect patterns that could indicate fraud or other risks. They can also be used to make automated decisions, such as approving loan applications or detecting suspicious transactions.

Blockchain technology: Blockchain technology can be used to enhance security and transparency in banking operations. It can be used to create tamper-proof records of transactions and customer information, which can improve the bank's compliance with regulatory requirements.

Cloud-based systems: Cloud-based systems can provide banks with increased scalability, flexibility, and cost savings. By moving their operations to the cloud, banks can reduce the need for expensive IT infrastructure and increase their ability to respond to changing business needs.

Digital banking channels: Banks can enhance the customer experience by providing more convenient digital banking channels, such as mobile banking and chatbots. This can help them reach a wider customer base and provide more personalized services.

Real-time analytics: Banks can use real-time analytics to monitor their operations and make data-driven decisions. By providing real-time visibility into key performance indicators, banks can quickly identify and address any issues that arise.

Biometrics: Biometrics can be used to authenticate customer identity, which can improve security and reduce the risk of fraud. This can be in form of facial recognition, fingerprint, voice and other biometric methods.

Open Banking: Open banking allows customers to share their financial data with third-party providers to access a wider range of services and products. Banks can leverage this to provide more personalized services, increase revenue and improve customer satisfaction.

All these future enhancements can be tailored to the specific needs of the bank and its customers and can help banks to improve their operations, reduce costs, and increase customer satisfaction.

References

[1] Fundamentals of database systems by (Elmasri Navathe, 2000)

Article: Student Information System: https://en.wikipedia.org/wiki/Student_information_system

[2] Learning MySQL : https://www.w3schools.com/MySQL/default.asp/

PHP Tutorials : https://www.javatpoint.com/php-tutorial/, http://www.youtube.com

[3] ER Diagram and Schema : https://erdplus.com

[4] Stack Overflow: https://stackoverflow.com/