INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

Online Banking System

PG-DAC March 2023

Submitted By:

Group No: 14

Roll No.	**Name:**
233024	**Sayali Bodre**
233021	**Kunal P. Bhangale**

Mrs.Sonali Mogal	**Mr. Rohit Puranik**
Project Guide	**Centre Coordinator**

Table of Contents

ABSTRACT. 03
ACKNOWLEDGEMENT. 05
INTRODUCTION. 06

FEATURES.. 06

1.1 PROJECT OBJECTIVE 06

1.2 PROJECT SCOPE. 06

SYSTEM REQUIREMENT SPECIFICATION 11

2.2 FUNCTIONAL REQUIREMENTS 12

2.3 NON FUNCTIONAL REQUIREMENT.. 13

SYSTEM DESIGN 15

3.1 INPUT AND OUTPUT DESIGN.. 15

3.1.1 INPUT DESIGN 15

3.1.2 OUTPUT DESIGN 15

DATABASE DESIGN.. 16

3.2 DATABASE 16

3.3 SYSTEM TOOLS.. 16

3.3.1 FRONT END.. 16

3.3.2 BACKEND 16

0 LEVEL DFD 17

1 LEVEL DFD 18

2 LEVEL DFD 19

PERSON.. 20

USE CASE DIAGRAM FOR ADMIN 21

USE CASE DIAGRAM FOR CUSTOMER 22

E-R DIAGRAM 24

E-R DIAGRAM MYSQL GENRATED.. 25

CLASS DIAGRAM 26

TABLE STRUCTURE.. 27

SCREENSHOTS.. 28

CONCLUSION ....

REFERENCES ....

LIST OF FIGURES

FIGURE 1:EMPLOYEE ACTIVITY DIAGRAM

FIGURE 2:CUSTOMER ACTIVITY DIAGRAM

FIGURE 3:0 LEVEL DFD

FIGURE 4:1 LEVEL DFD

FIGURE 5:2 LEVEL DFD

FIGURE 6:USE CASE DIAGRAM FOR EMPLOYEE

FIGURE 7:USE CASE DIAGRAM FOR CUSTOMER

FIGURE 8: ENTITY REALTION DIAGRAM

FIGURE 9: ER(WORKBENCH GENREATED)

ABSTRACT

The Banking Management System project aimed to create a functional banking application for learning purposes. Covering customer interactions, employee management, account administration, transactions, and bank branches, the agile approach mirrored real-world software development. Challenges included data validation, secure transactions, and module integration.

The report details project goals, execution, challenges, achievements, and the lessons learned. By sharing insights, the project contributes to the understanding of software development and project management, offering a foundation for future endeavors and enriching the skills of aspiring developers in the rapidly evolving financial technology landscape.

Challenges included data validation and secure transactions, fostering critical thinking and problem-solving. This concise report provides an overview of the project's scope, challenges, achievements, and lessons learned, contributing insights to software development and project execution. The project emphasizes not just the application, but the journey of growth and skill acquisition.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who has contributed to the completion of our project.

First and foremost, We would like to thank our project guide **Mrs.Sonali Ma'am** for their constant guidance and support throughout the project. We extend our sincere thanks to our respected Centre Co-Ordinator **Mr. Rohit Puranik**, for allowing us to use the facilities available.

We would also like to express our appreciation to the faculty members of our department for their constructive feedback and encouragement. Their insights and suggestions have helped us to refine our ideas and enhance the quality of our work.

Furthermore, we would like to thank our families and friends for their unwavering support and encouragement throughout our academic journey. Their love and support have been a constant source of motivation and inspiration for us.

Thank you all for your valuable contributions to our project.

Sayali Bodre (233024)

Kunal P. Bhangale (233024)

INTRODUCTION

The Banking Management System is a contemporary online platform revolutionizing the retail banking landscape. Traditionally, acquiring financial services and products involved multiple intermediaries, limiting profitability for retailers. In response, this system establishes a direct connection between customers, employees, accounts, transactions, and bank branches. In the past, retailers often contended with narrow margins due to multi-tier distribution networks. This platform disrupts the status quo by facilitating direct interactions between customers and financial services, enabling retailers to secure products from bank branches with enhanced ease. The integration of customer, employee, account, transaction, and bank branch functionalities empowers efficient, user-friendly experiences.

By consolidating customer interactions, employee management, account administration, transaction processing, and bank branch operations into one streamlined digital platform, the Banking Management System enables swift and seamless business operations.

Features: -

1. **Customer Management**: Implementing features for customer registration, profile management, and personal information updates.
2. **Employee Management:** Enabling the management of bank staff, their roles, permissions, and associated data.
3. **Account Administration**: Developing functionalities for opening new accounts, updating account details, and managing account types.
4. **Transaction Processing**: Designing mechanisms for conducting various types of transactions, ensuring data accuracy and security.
5. **Bank Branches**: Including modules to manage multiple bank branches, their details, and associated activities.

1.1 PROJECT OBJECTIVE

The objective of the Banking Management System project was two fold: to design and implement a functional banking application and to provide an immersive learning platform for students to grasp key concepts of software development, project management, and collaboration. As technology continues to redefine the way financial institutions operate, the importance of equipping future professionals with practical skills in software development cannot be overstated.

1.2 PROJECT SCOPE

The Banking Management System project encompasses the design and development of a comprehensive software solution to streamline banking operations. This includes facilitating customer interactions, managing employee roles, administering accounts, processing transactions, and overseeing bank branch activities. The scope further extends to creating a user-friendly interface for customers and employees, ensuring secure and accurate transaction processing, and enabling efficient management of account details and balances. By centralizing these functionalities into a cohesive system, the project aims to enhance user experiences, improve operational efficiency, and provide administrators with robust oversight capabilities.

Activity Diagrams:

Activity Diagram For Employee

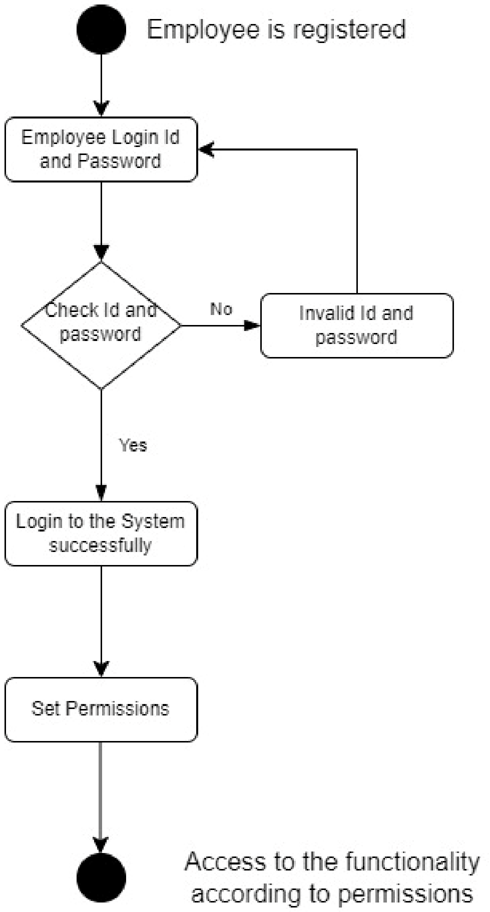

Employee is registered
Employee Login Id and Password
Check Id and password
No
Invalid Id and password
Yes
Login to the System successfully
Set Permissions
Access to the functionality according to permissions

Figure 1 Admin Activity Diagram

Activity diagram customer

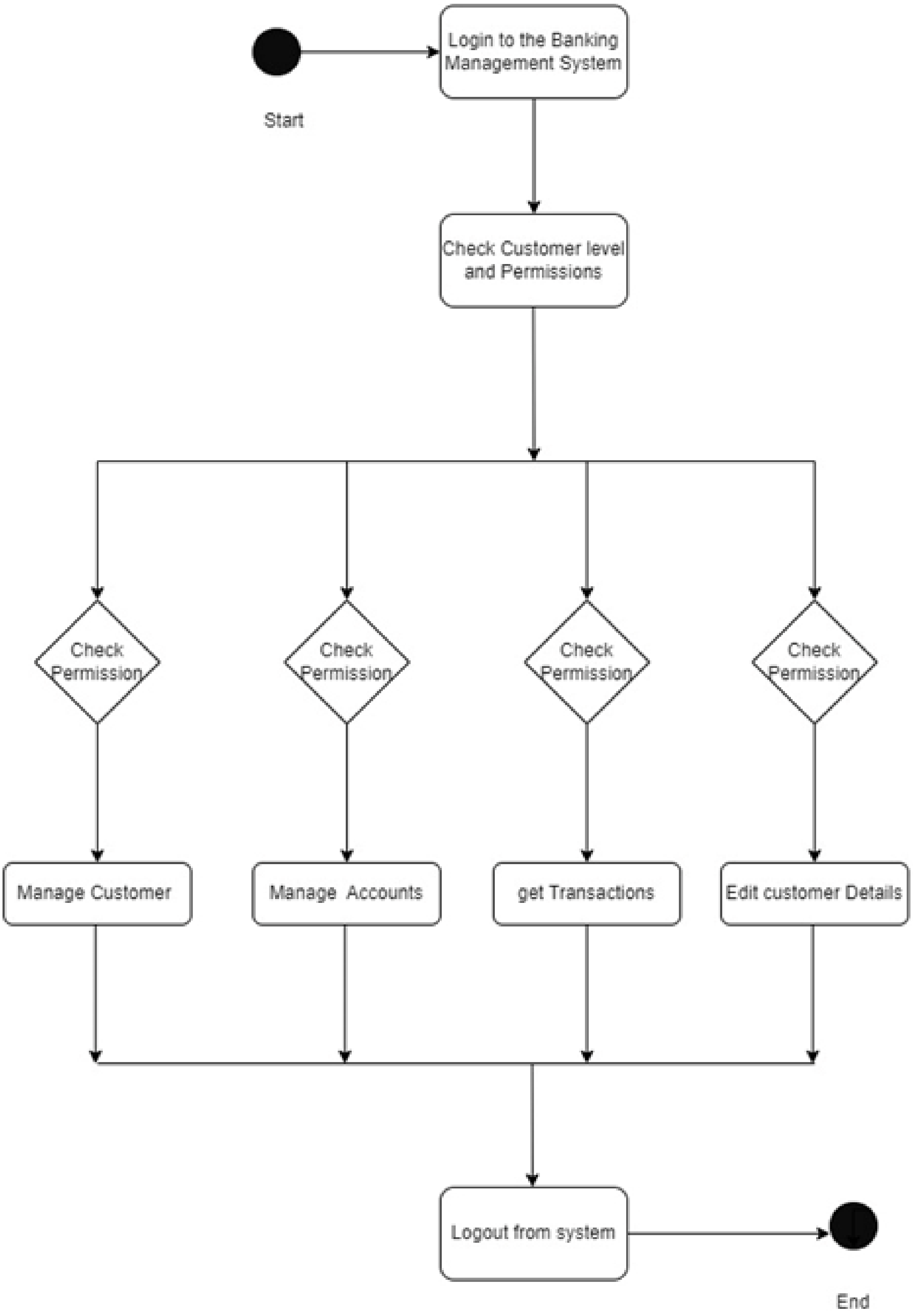


Figure 2 Customer Activity Diagram

2.System Requirement specification

Title:

Software Requirements Specification for Online e-Banking system.

Team:

Customer,Employee

Objective:

The objective of the Banking Management System project was two fold: to design and implement a functional banking application and to provide an immersive learning platform for students to grasp key concepts of software development, project management, and collaboration. As technology continues to redefine the way financial institutions operate, the importance of equipping future professionals with practical skills in software development cannot be overstated.

Scope:

The software will encompass the following key modules:

Employee Management

- Administrators can create, modify, and delete employee records.
- Employee profiles consist of unique identification numbers, roles, names, contact information, and branch assignments.
- Authentication and authorization mechanisms restrict unauthorized access to employee data.

Customer Management

- Employees can create, update, and manage customer profiles.
- Customer information includes unique identification numbers, personal details, contact information, and linked accounts.

Account Management

- Support for various account types (e.g., savings, checking, loans).
- Each account links to a customer and a specific branch.
- Accurate recording and retrieval of account balances, transaction history, and statuses.

Transaction Management

- Customers and employees can initiate transactions: deposits, withdrawals, transfers, loan payments.
- Transaction records capture timestamps, types, and involved parties.
- Stringent validation prevents erroneous and unauthorized transactions.

Branch Management

- Functionality for adding, updating, and removing branches.
- Each branch has a unique identifier, name, address, and associated employees.
- Tracking of branch-specific data, including transaction volumes and account statistics.

System Overview:

The Online E-Banking System empowers users to:

- Manage multiple bank accounts, including savings and current accounts.
- Initiate secure fund transfers between accounts.
- View real-time transaction history and account balances.
- Pay bills, including utility bills and credit card bills.
- Apply for loans, monitor loan applications, and make loan payments.
- Receive real-time notifications for important account activities.
- Access customer support and assistance.

2.1 Functional Requirements

Employee Management

- **Create Employee:** Administrators can add employee records with roles, names, contact details, and branch assignments.
- **Update Employee:** Employee information can be modified, including role changes and contact details.
- **Delete Employee:** Administrators can remove employee records from the system.
- **Access Control:** Different user roles (administrator, manager, employee) have varying levels of access to employee data.

Customer Management

- **Create Customer**: Employees can create customer profiles with personal details and contact information.
- **Update Customer**: Customer data can be modified, such as contact details and personal information.
- **Manage Accounts:** Employees can link customer profiles to relevant accounts.
- **Access Control**: Only authorized employees can create and manage customer records.

Account Management

- **Account Creation**: Employees can create different types of accounts, associating them with customers and branches.
- **Account Status**: Account statuses include active, inactive, and closed.
- **Balance Tracking**: Real-time tracking of account balances, with updates after each transaction.
- **Account Closure**: Accounts can be closed by employees upon customer request or due to inactivity.

Transaction Management

- **Initiate Transactions**: Customers and employees can initiate transactions, such as deposits, withdrawals, transfers, and loan payments.
- **Transaction Recording**: Each transaction is recorded with a timestamp, type, parties involved, and amount.
- **Transaction Validation**: System validates transaction details, including sufficient funds and account validity.

Branch Management

- **Add Branch**: Administrators can add new branches, providing unique identifiers, names, addresses, and employee assignments.
- **Update Branch**: Branch information can be updated, such as address changes and employee assignments.
- **Remove Branch**: Administrators can remove branches after ensuring all necessary data is transferred.
- **Branch Statistics**: System tracks branch-specific data, like transaction volumes and account growth.

2.2 Non-Functional Requirements

Security

- **User Authentication**: Secure login mechanism for employees with role-based access.
- **Data Privacy**: Customer and employee data is encrypted and protected from unauthorized access.
- **Access Auditing**: Maintain logs of user activities, including login attempts and data modifications.

Performance

- **Scalability**: The system should handle increased load without significant performance degradation.
- **Response Time**: User interactions and data retrieval should have minimal delays.

Reliability

- **High Availability**: Aim for 99.9% system uptime to ensure consistent service availability.
- **Backup and Recovery**: Regular data backups and robust disaster recovery procedures.

User Interface

- **Intuitive UI**: User-friendly interface design for easy navigation and efficient task execution.
- **Device Compatibility**: Responsive design to support access from various devices (desktop, mobile).

Reporting and Analytics

- **Reporting Dashboard**: Provide users with insights into account activities, transaction trends, and branch performance.
- **Data Visualization:** Present data through graphs, charts, and tables for easier understanding.

Maintainability

- A Commercial database software will be used to maintain System data Persistence.
- IT operations team will easily monitor and configure System using administrativetools provided by Servers.

Efficiency:

- On peak time, maximum number of users will place order, view products with sameresponse time.
- System will be able to manage all transactions with isolation.

Scalability:

- System will be able to provide consistent user experience to stake holder as well asvisitors irrespective of load.

Safety:

- Online portal functionalities are protected from outside with proper firewallconfiguration.
- Online portal will be always kept updated with latest antivirus software.
- Business data will be backed up periodically to ensure safety of data usingincremental back up strategy.

3.SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design. Specifications to performance specification. System design has two phases of development.

- Logical Design
- Physical Design

During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

3.1 INPUT AND OUTPUT DESIGN

3.1.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

DATABASE DESIGN

3.2 DATABASE

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

▪ Primary key - the field that is unique for all the record occurrences

▪ Foreign key - the field used to set relation between tables Normalization is a technique to avoid redundancy in the tables.

3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

3.3.1 FRONT END:

React is a library which is developed by Facebook are utilized to implement the frontend. React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Version : **"react": "^17.0.2".**

3.3.2 BACKEND:

Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

Version Used : **2.7.14**

3.3.3 Database:

MySQL is used to design databases.

MySQL:
MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language.
Version : **8.0.32**

DFD Diagrams:

0 LEVEL DFD

F

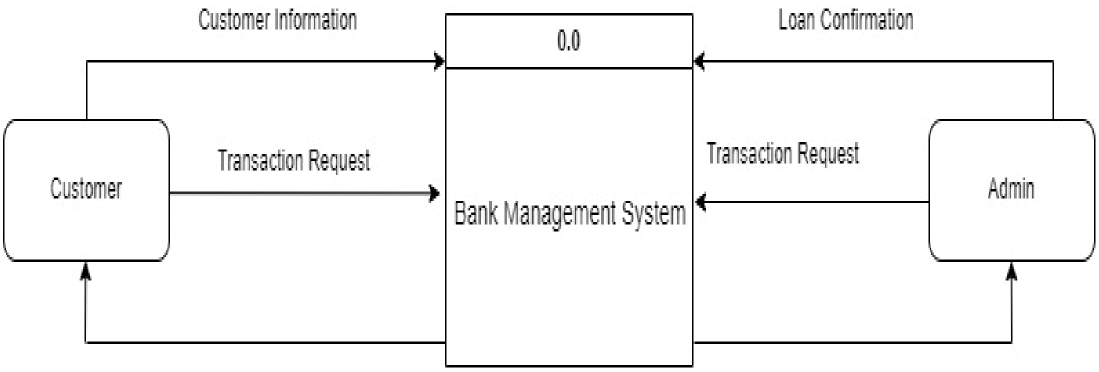


Figure 3 : 0 Level DFD Diagram

1 LEVEL DFD DIAGRAM

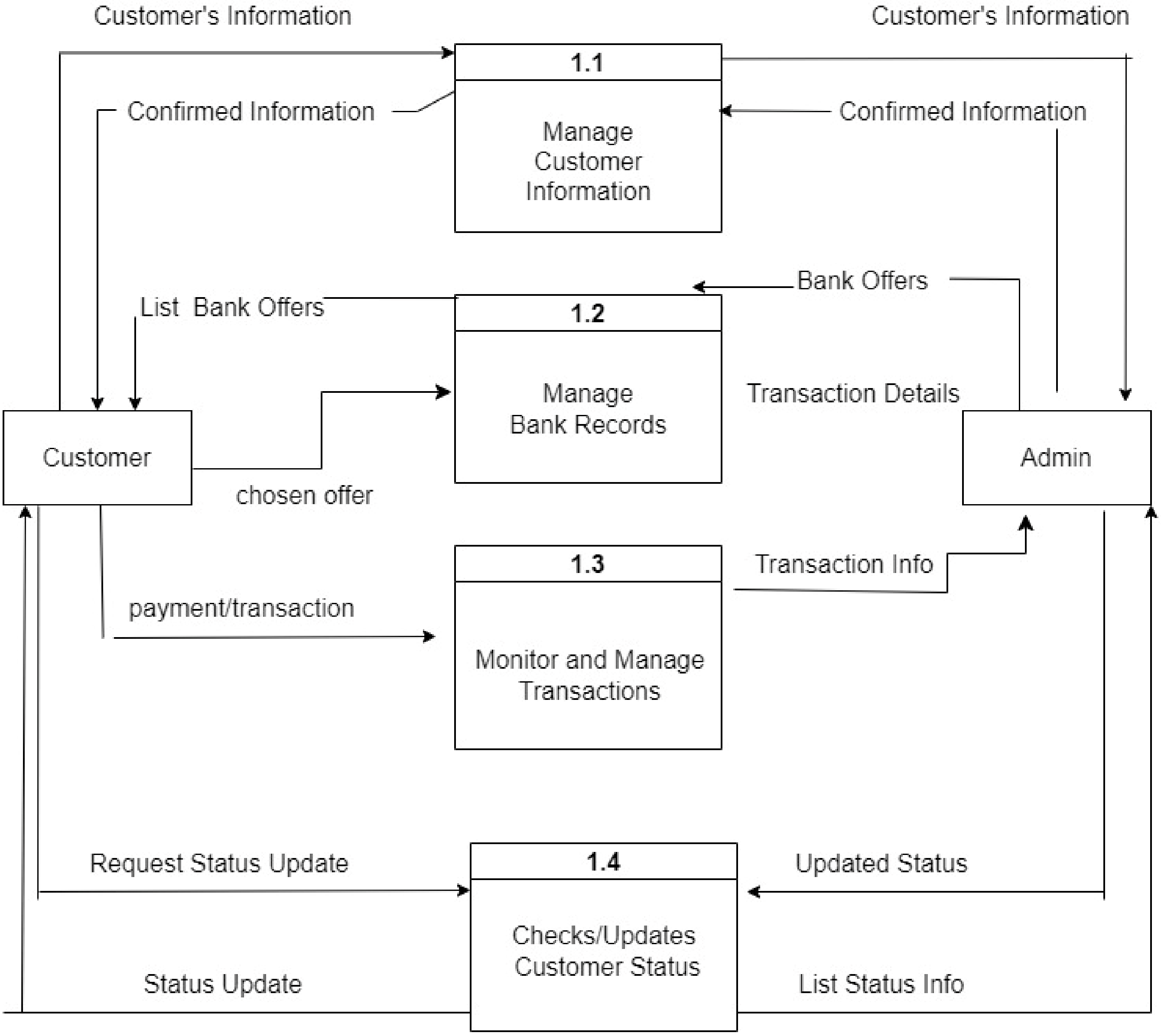


Figure 4 LEVEL 1 DFD Diagram

2 LEVEL DFD DIAGRAM

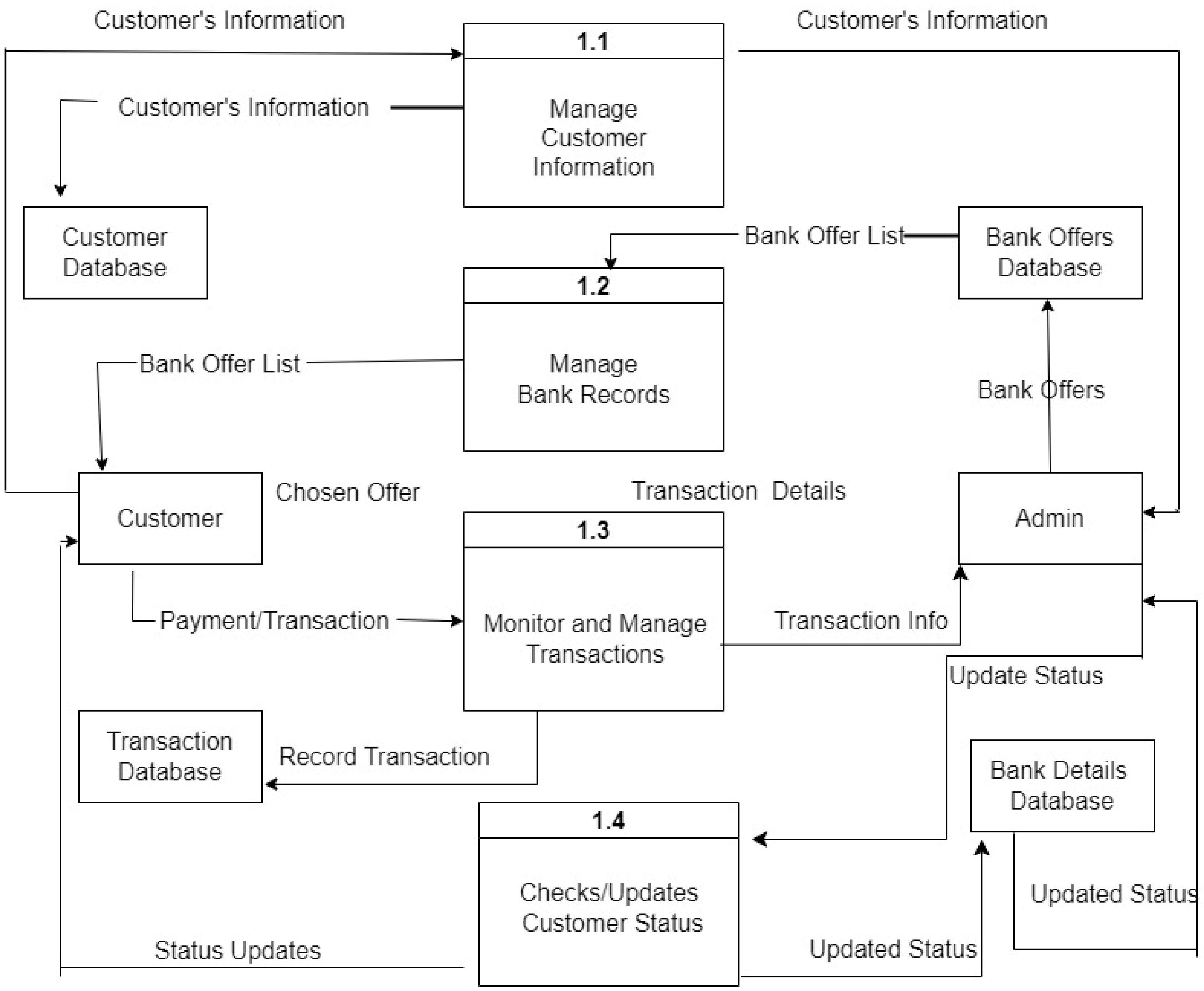


Figure 5 :LEVEL 2 DFD Diagram

Use Case Diagram

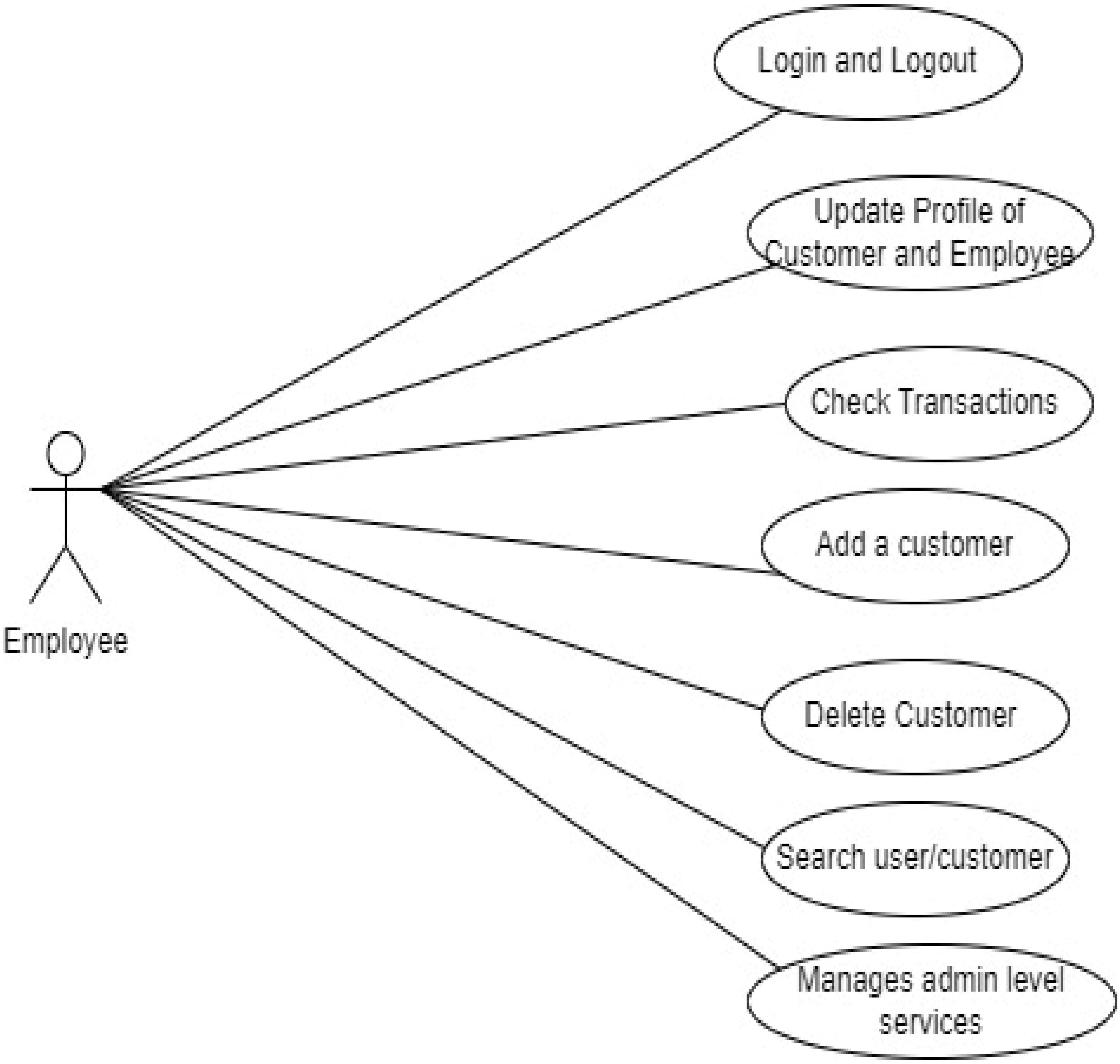


Figure 6 Employee Use Case Diagram

Customer Usecase :

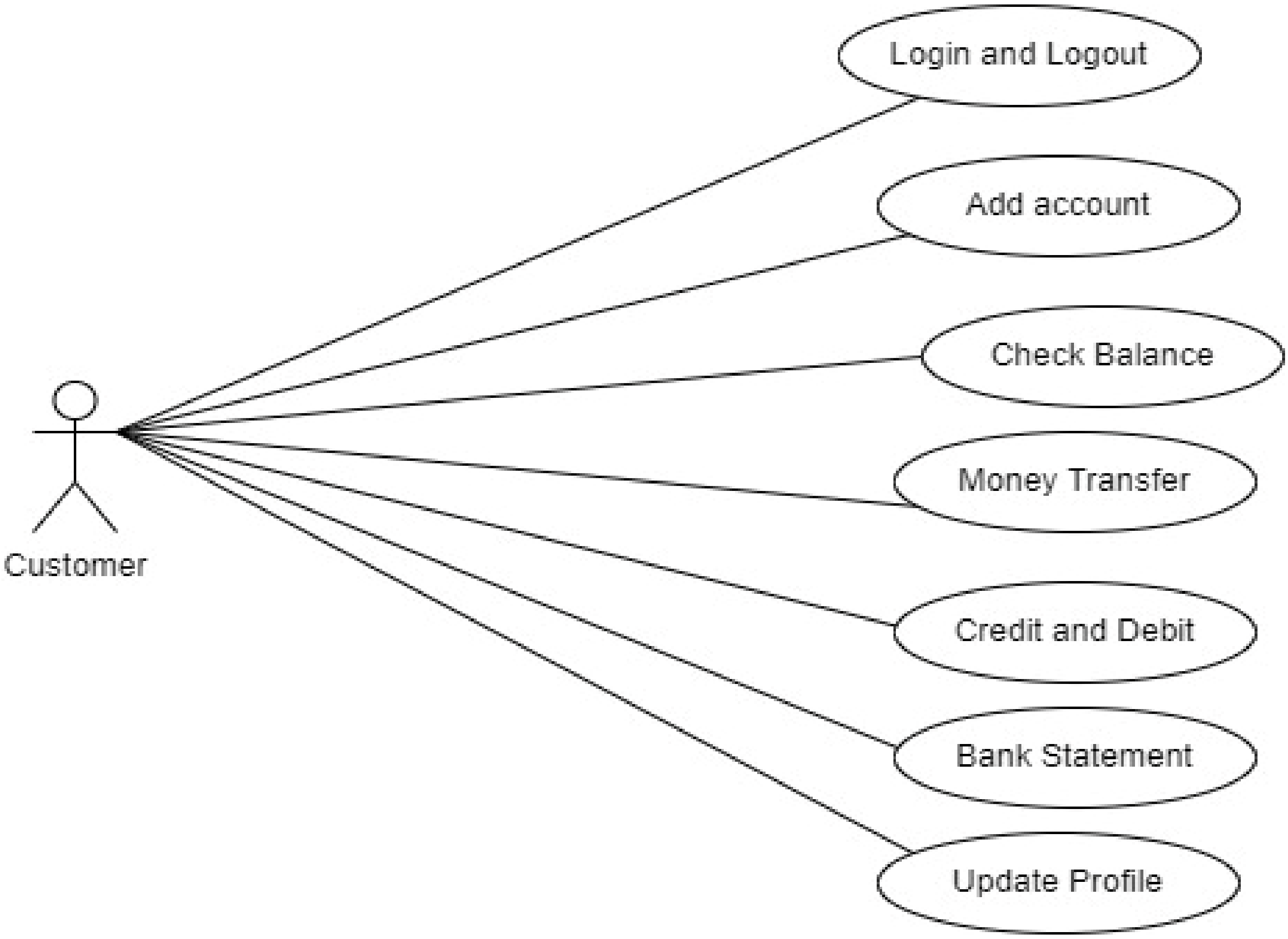


Figure 7 CustomerUse Case Diagram

E-R Diagram:

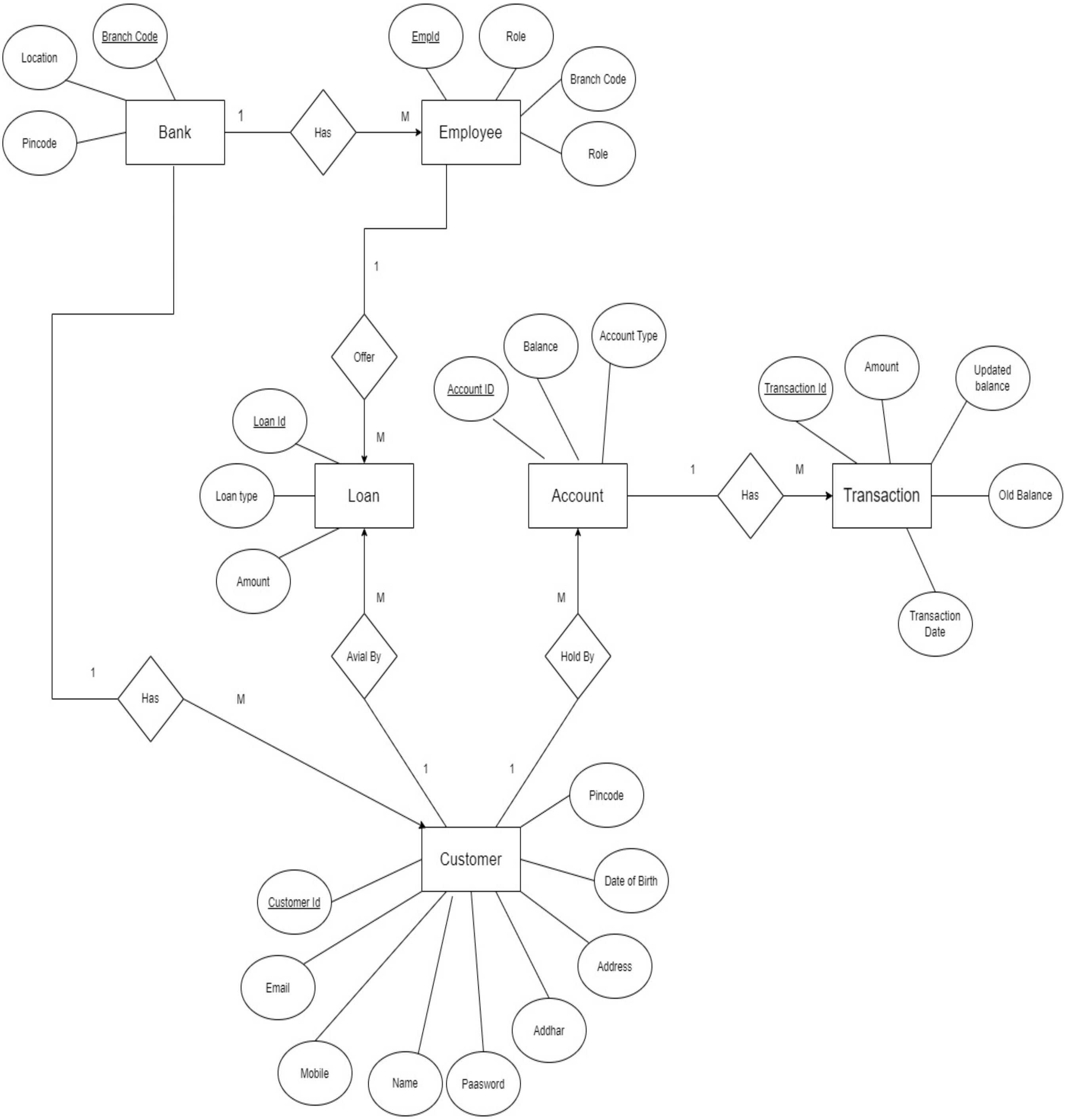


Figure 8 ER Diagram

E-R Diagram:

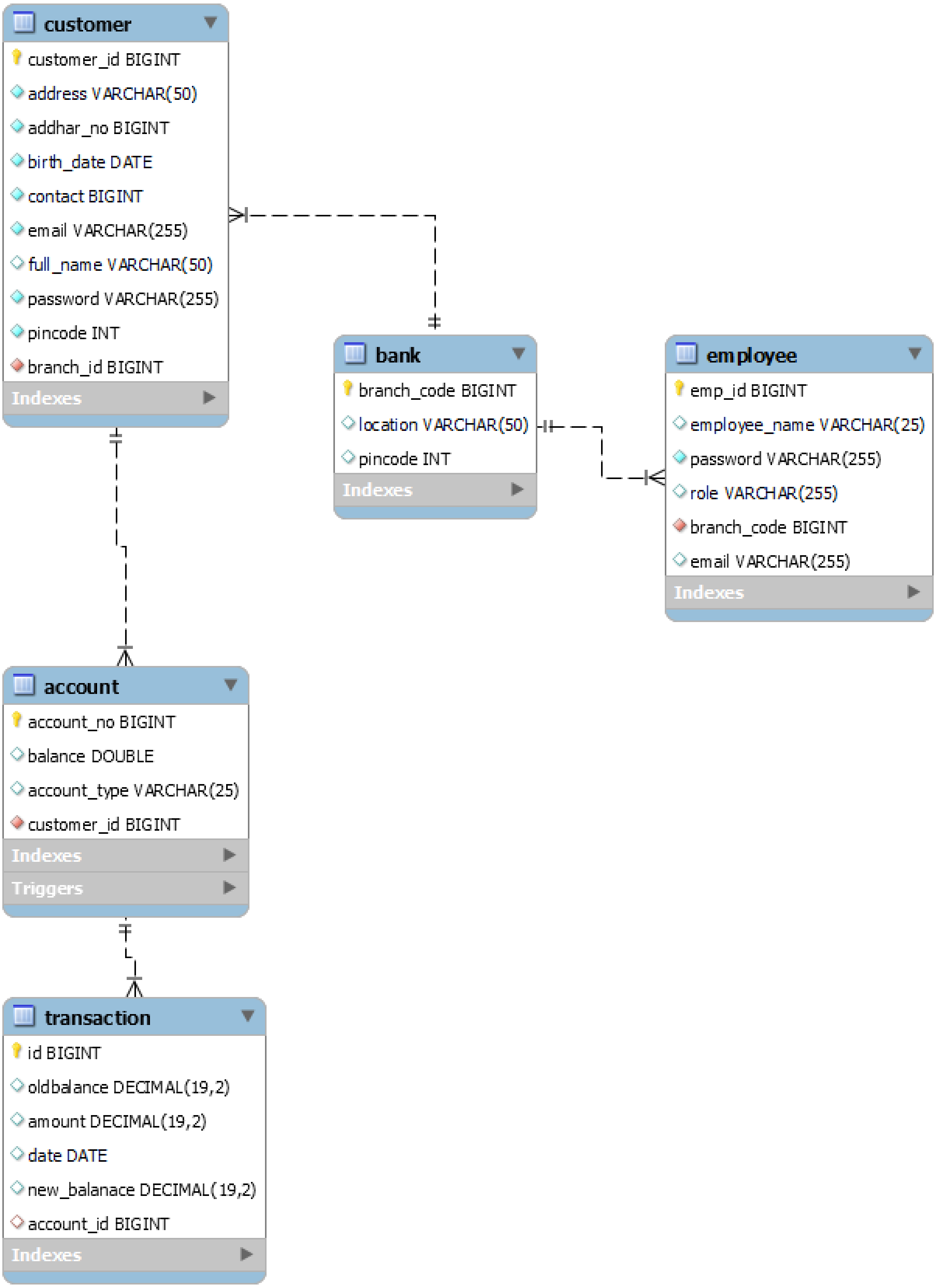
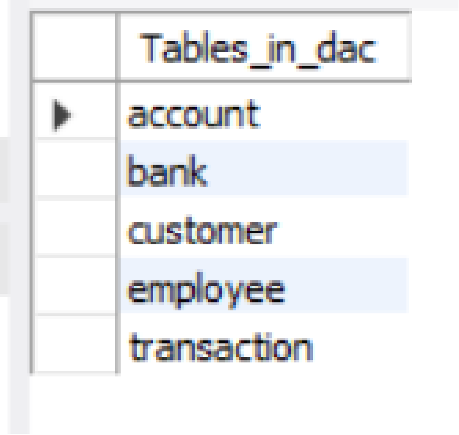


Figure 9 ER Diagram(MY Sql Genrated)

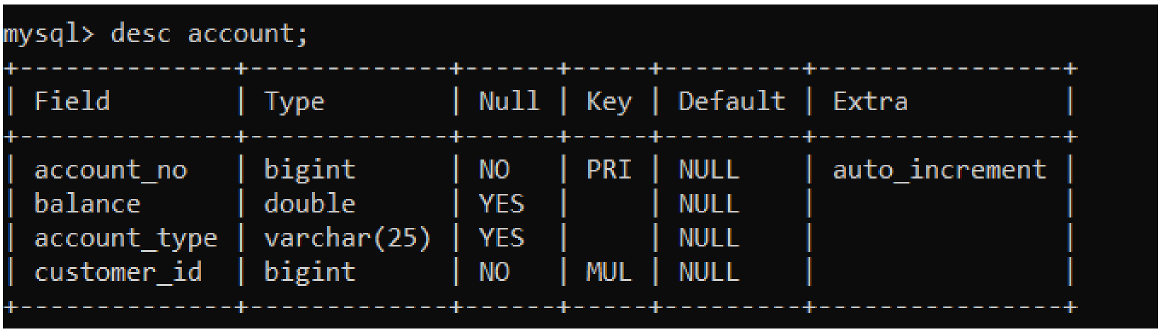
TABLE STRUCTURE:

Tables:



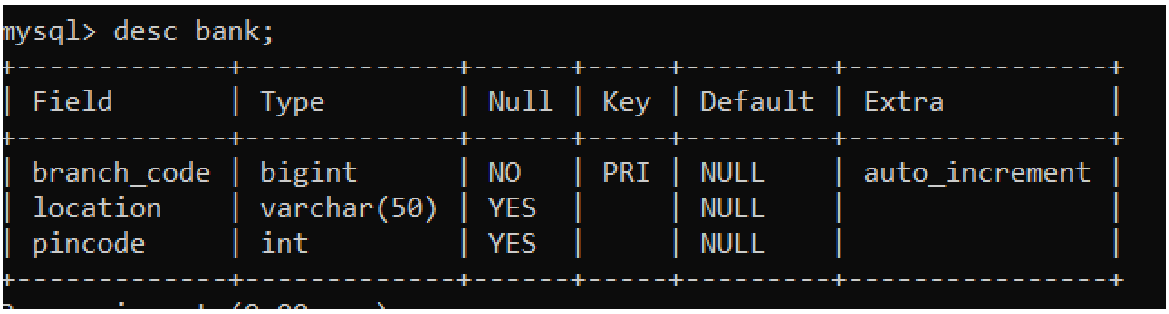
Tables_in_dac
account
bank
customer
employee
transaction

Account:



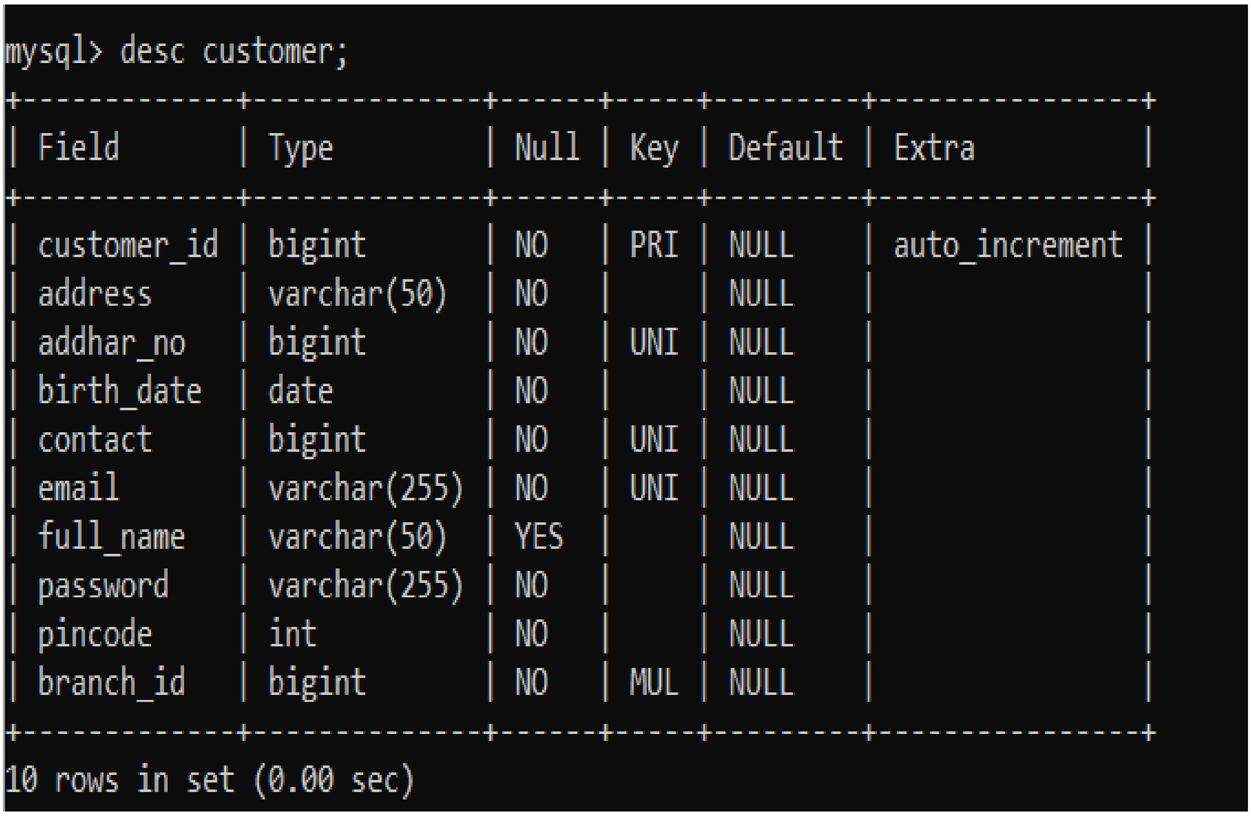
```
mysql> desc account;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| account_no   | bigint      | NO   | PRI | NULL    | auto_increment |
| balance      | double      | YES  |     | NULL    |                |
| account_type | varchar(25) | YES  |     | NULL    |                |
| customer_id  | bigint      | NO   | MUL | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
```

Bank:



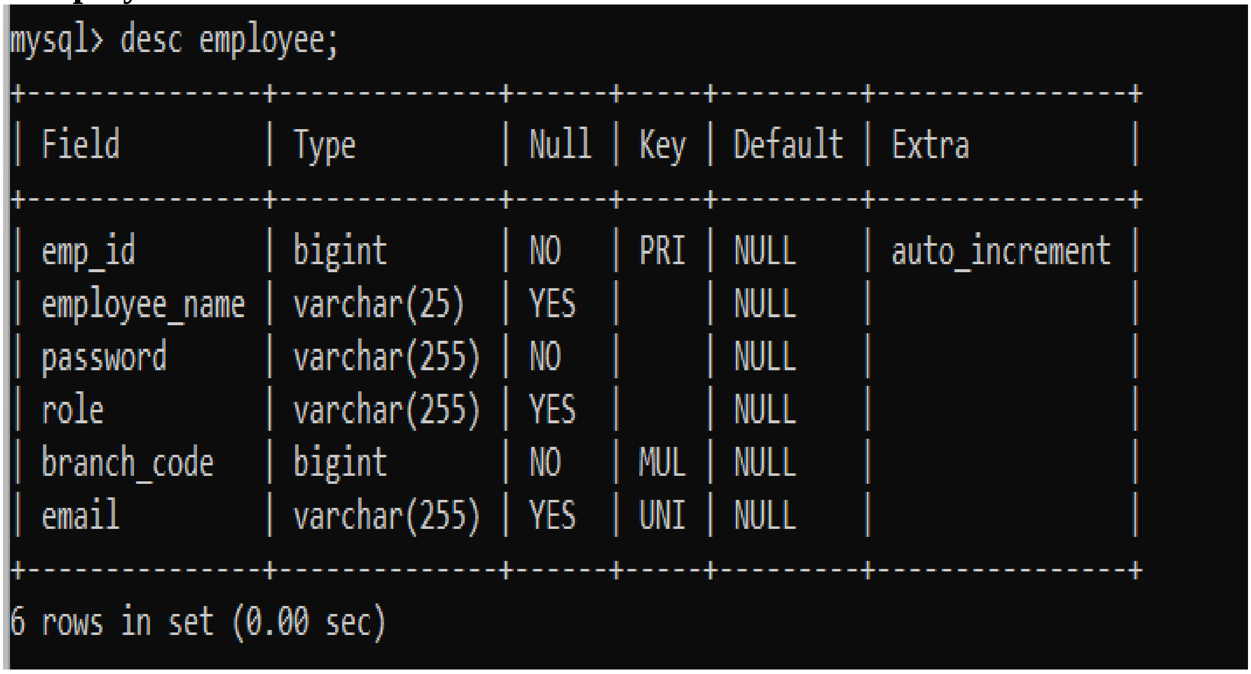
```
mysql> desc bank;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| branch_code | bigint      | NO   | PRI | NULL    | auto_increment |
| location    | varchar(50) | YES  |     | NULL    |                |
| pincode     | int         | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
```

Customer :



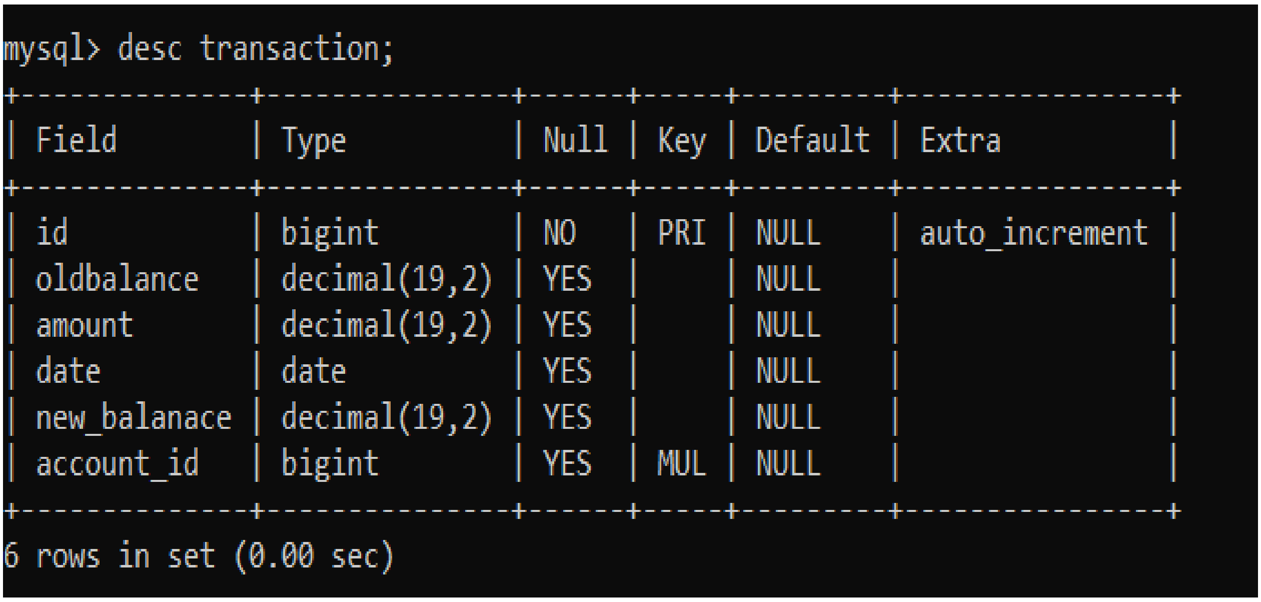
```
mysql> desc customer;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| customer_id | bigint       | NO   | PRI | NULL    | auto_increment |
| address     | varchar(50)  | NO   |     | NULL    |                |
| addhar_no   | bigint       | NO   | UNI | NULL    |                |
| birth_date  | date         | NO   |     | NULL    |                |
| contact     | bigint       | NO   | UNI | NULL    |                |
| email       | varchar(255) | NO   | UNI | NULL    |                |
| full_name   | varchar(50)  | YES  |     | NULL    |                |
| password    | varchar(255) | NO   |     | NULL    |                |
| pincode     | int          | NO   |     | NULL    |                |
| branch_id   | bigint       | NO   | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)
```

Employee:



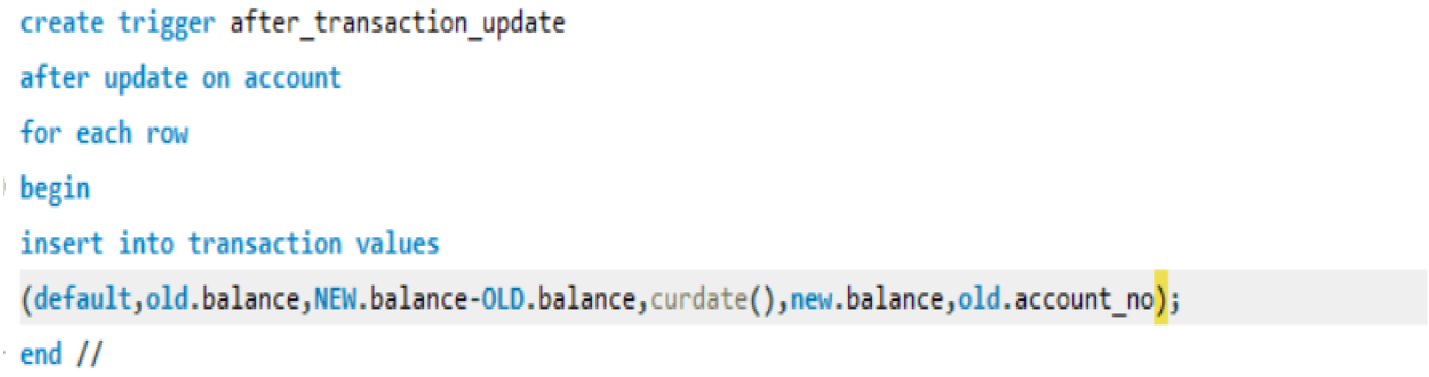
```
mysql> desc employee;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| emp_id        | bigint       | NO   | PRI | NULL    | auto_increment |
| employee_name | varchar(25)  | YES  |     | NULL    |                |
| password      | varchar(255) | NO   |     | NULL    |                |
| role          | varchar(255) | YES  |     | NULL    |                |
| branch_code   | bigint       | NO   | MUL | NULL    |                |
| email         | varchar(255) | YES  | UNI | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Transaction:

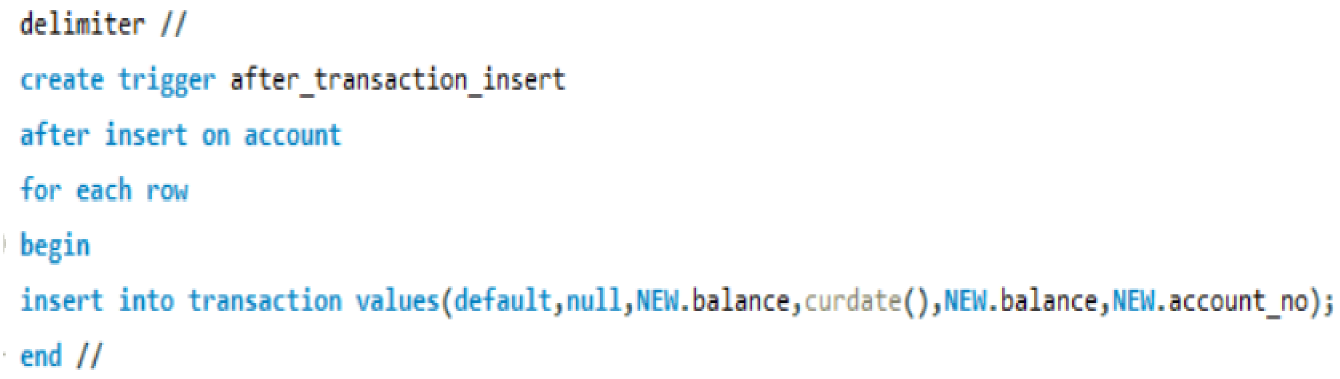


```
mysql> desc transaction;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| id           | bigint        | NO   | PRI | NULL    | auto_increment |
| oldbalance   | decimal(19,2) | YES  |     | NULL    |                |
| amount       | decimal(19,2) | YES  |     | NULL    |                |
| date         | date          | YES  |     | NULL    |                |
| new_balanace | decimal(19,2) | YES  |     | NULL    |                |
| account_id   | bigint        | YES  | MUL | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Triggers used for Maintaining transactions



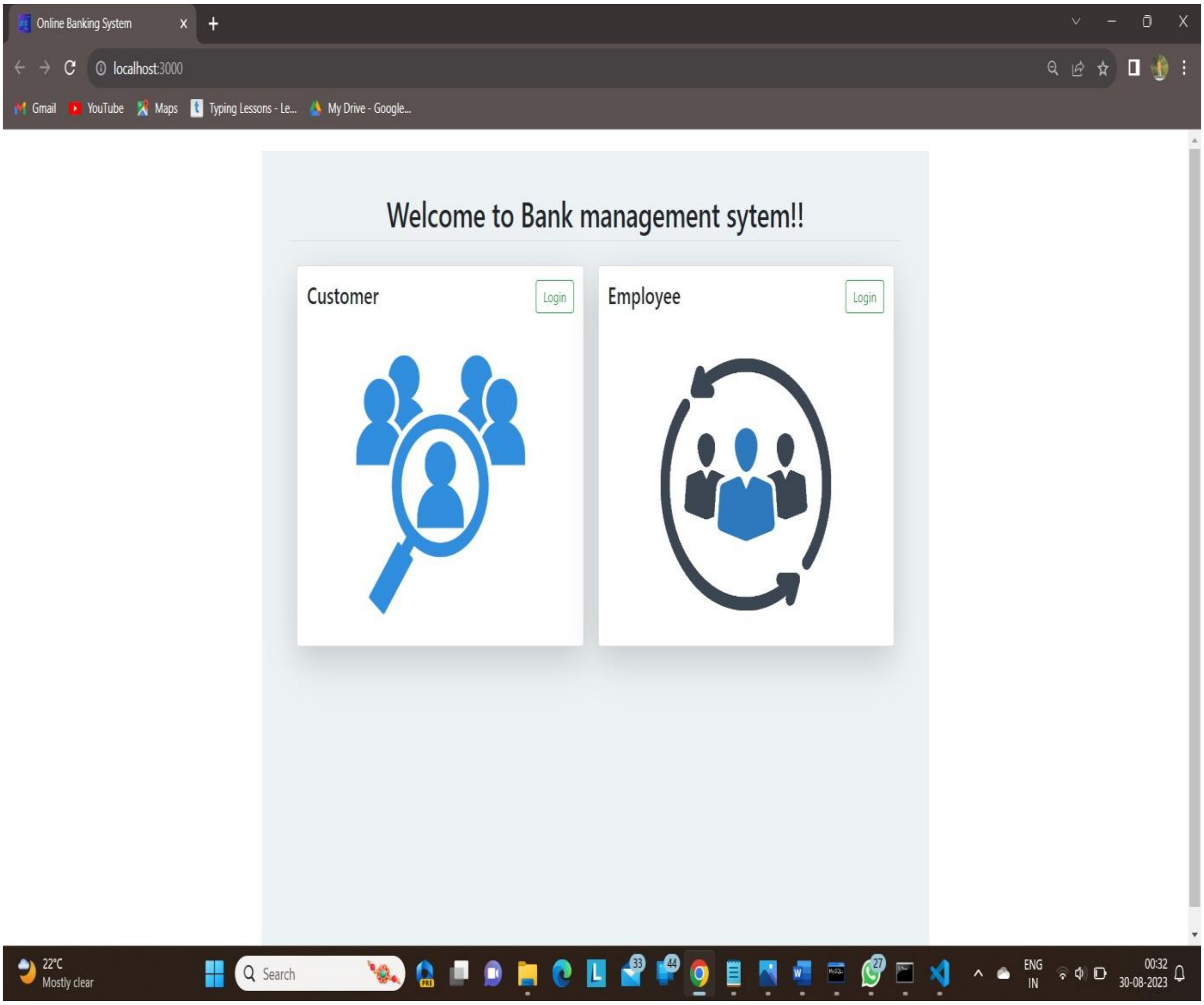
```
create trigger after_transaction_update
after update on account
for each row
begin
insert into transaction values
(default,old.balance,NEW.balance-OLD.balance,curdate(),new.balance,old.account_no);
end //
```



```
delimiter //
create trigger after_transaction_insert
after insert on account
for each row
begin
insert into transaction values(default,null,NEW.balance,curdate(),NEW.balance,NEW.account_no);
end //
```

Screenshots :

HomePage/LoginPage :

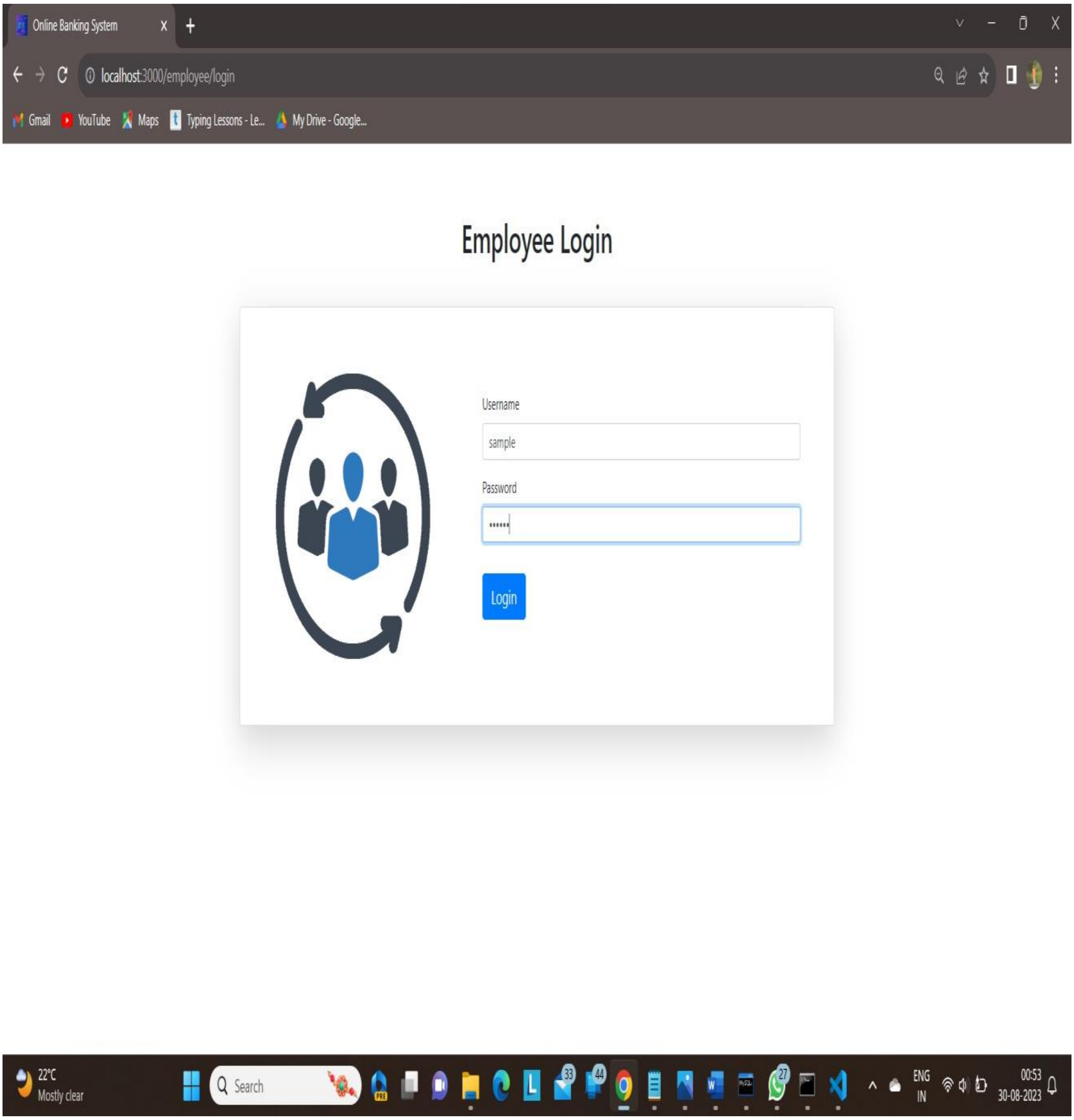


Welcome to Bank management sytem!!

Customer Login

Employee Login

Employee SignIn :



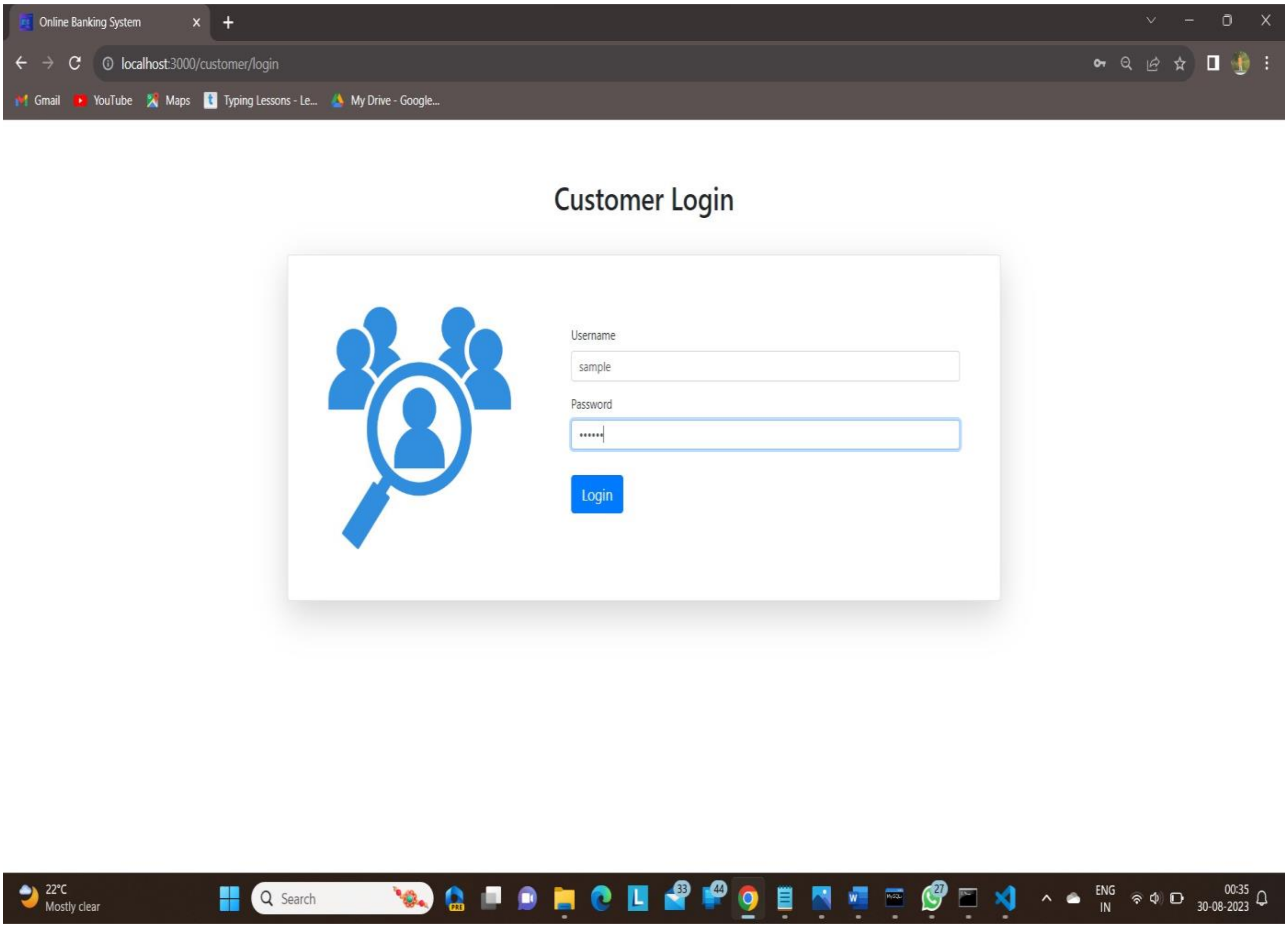
Username

sample

Password

••••••

Customer SignIn :



Customer Login

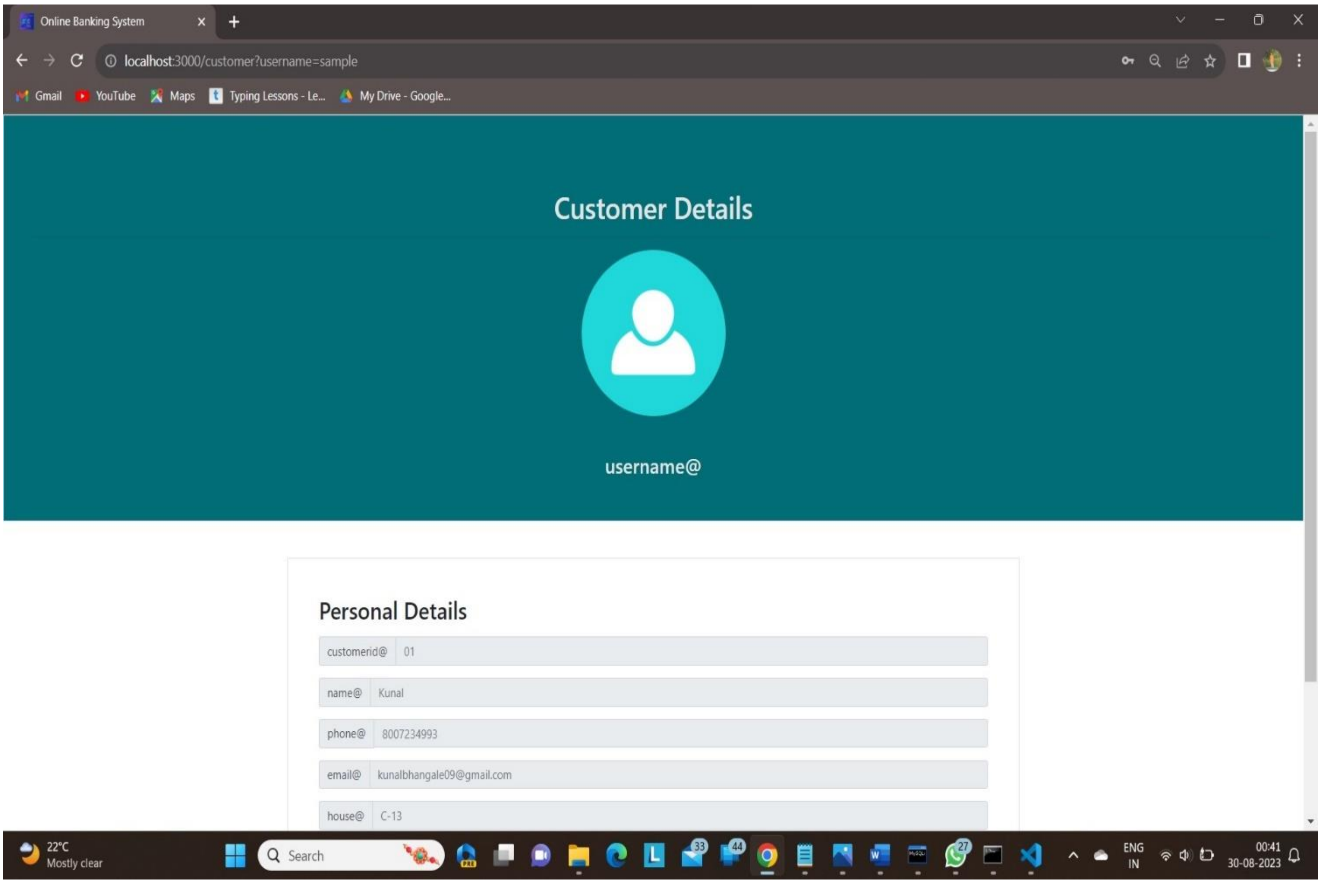
Username

sample

Password

••••••

Customer details :



Customer Details

username@

Personal Details

customerid@ 01

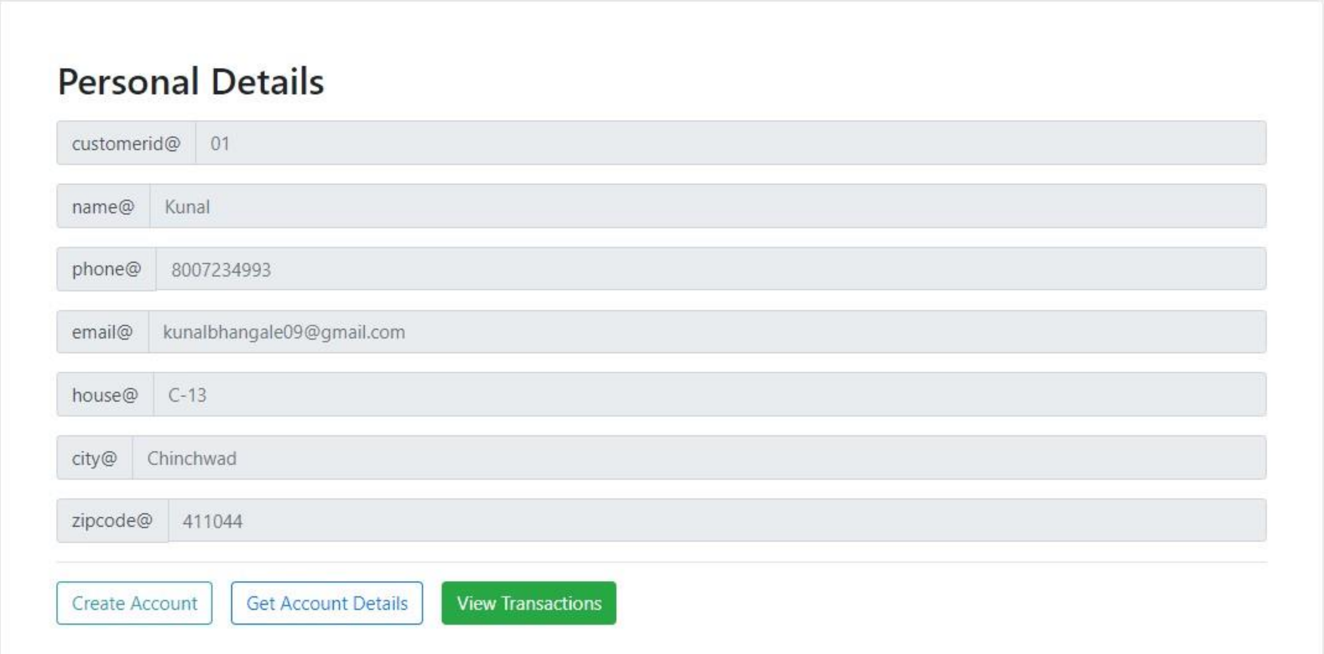
name@ Kunal

phone@ 8007234993

email@ kunalbhangale09@gmail.com

house@ C-13

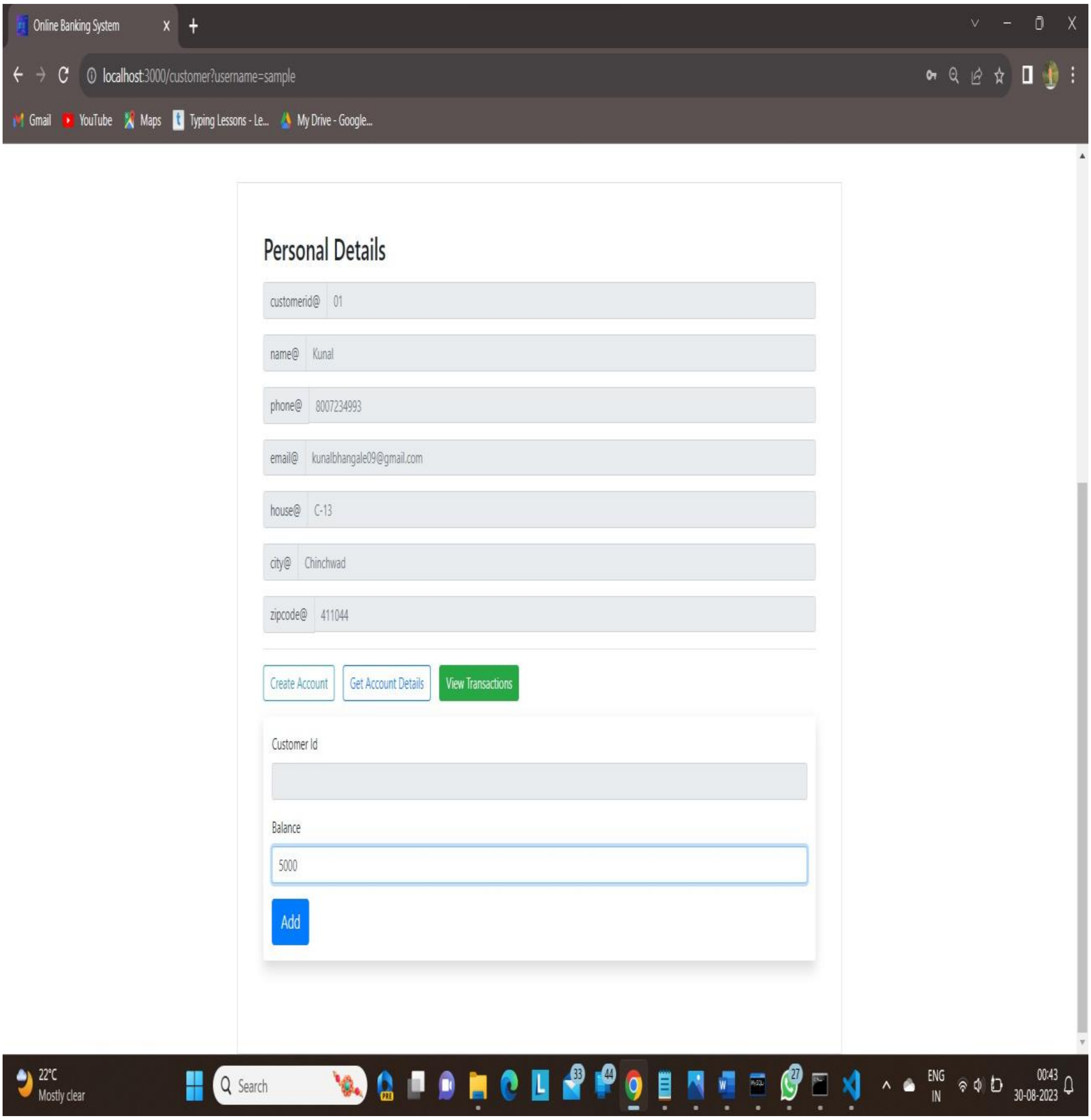
Personal details :



Personal Details

customerid@	01
name@	Kunal
phone@	8007234993
email@	kunalbhangale09@gmail.com
house@	C-13
city@	Chinchwad
zipcode@	411044

Add account :



Personal Details

customerid@ 01

name@ Kunal

phone@ 8007234993

email@ kunalbhangale09@gmail.com

house@ C-13

city@ Chinchwad

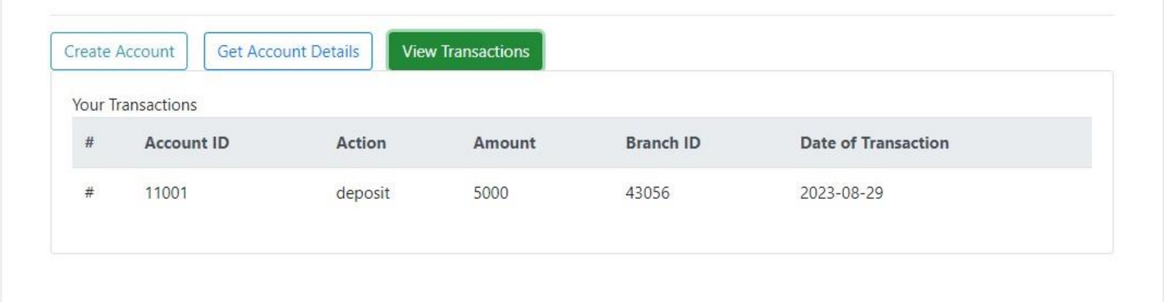
zipcode@ 411044

Customer Id

Balance

5000

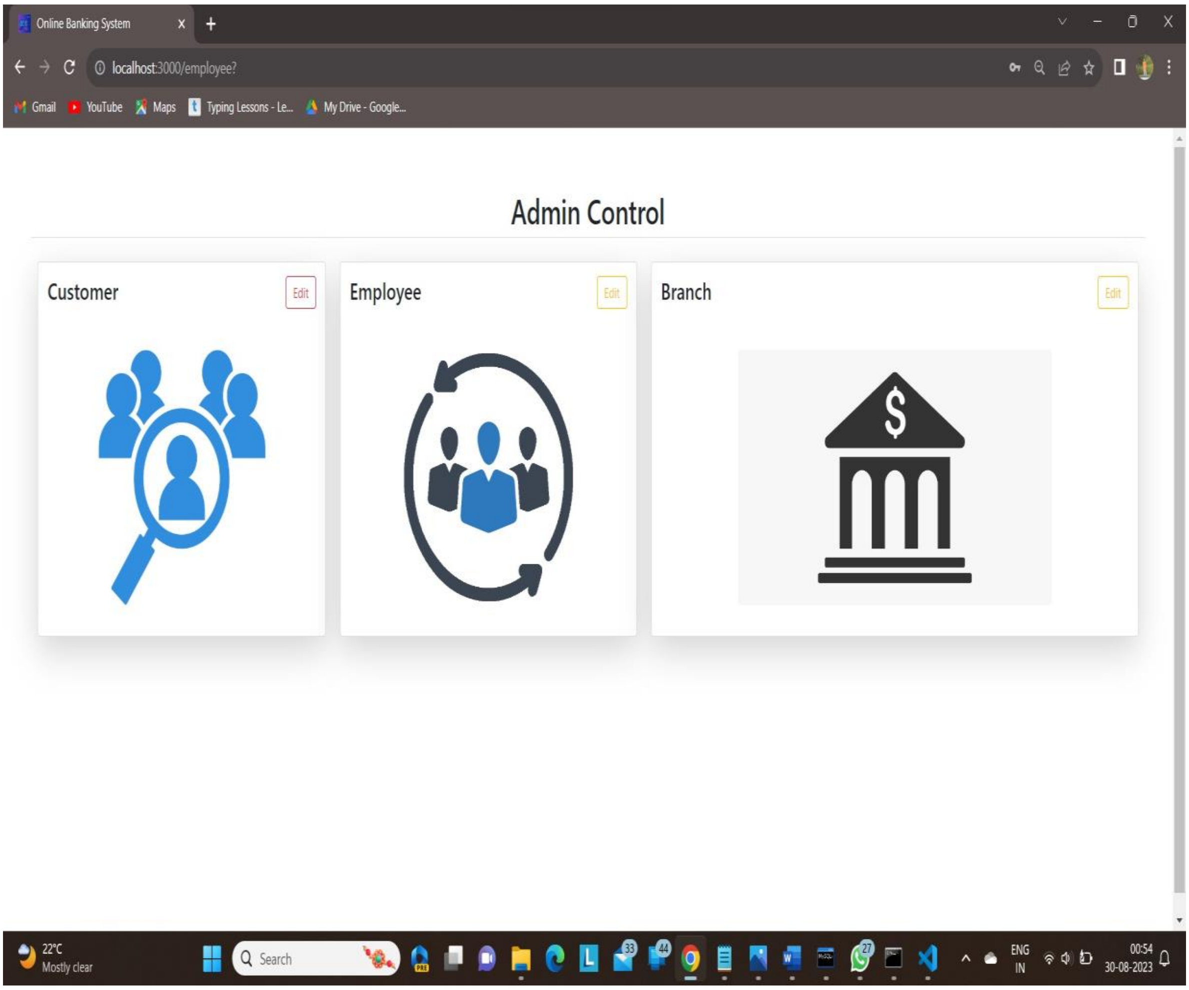
Transaction List :



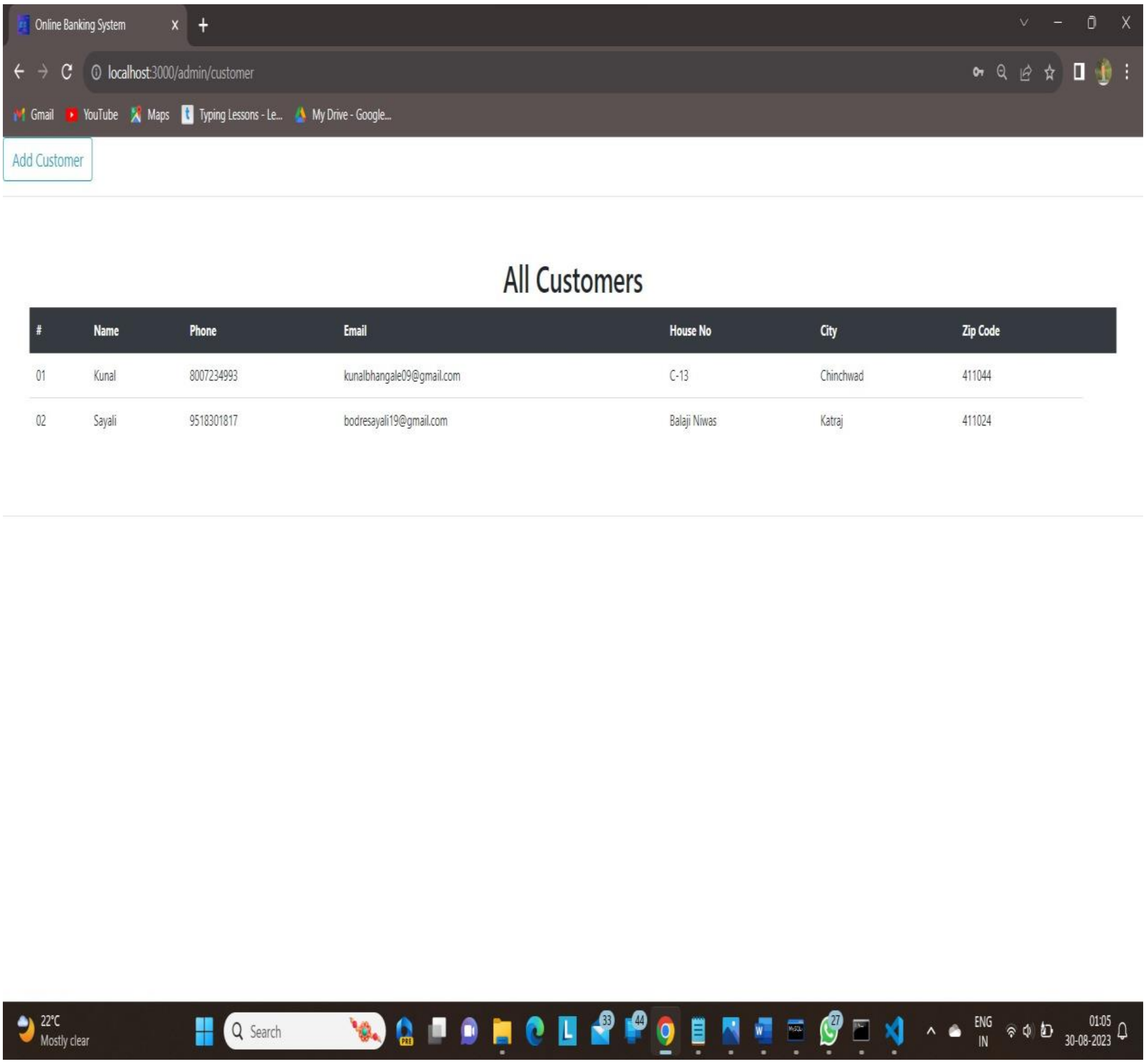
Your Transactions

#	Account ID	Action	Amount	Branch ID	Date of Transaction
#	11001	deposit	5000	43056	2023-08-29

Employee Dashboard :



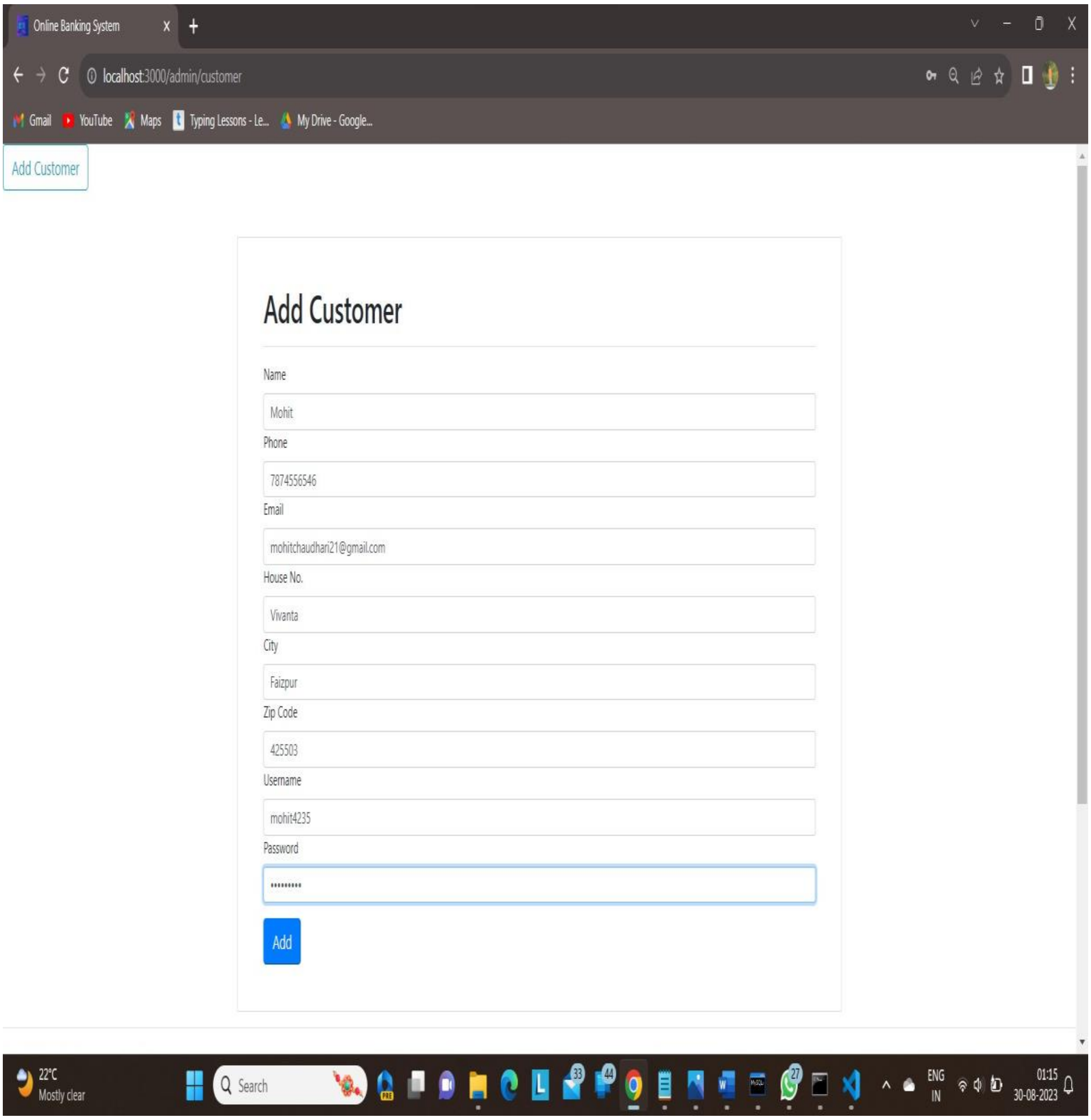
Customer List :



All Customers

#	Name	Phone	Email	House No	City	Zip Code
01	Kunal	8007234993	kunalbhangale09@gmail.com	C-13	Chinchwad	411044
02	Sayali	9518301817	bodresayali19@gmail.com	Balaji Niwas	Katraj	411024

Add Customer :



Add Customer

Name: Mohit

Phone: 7874556546

Email: mohitchaudhari21@gmail.com

House No.: Vivanta

City: Faizpur

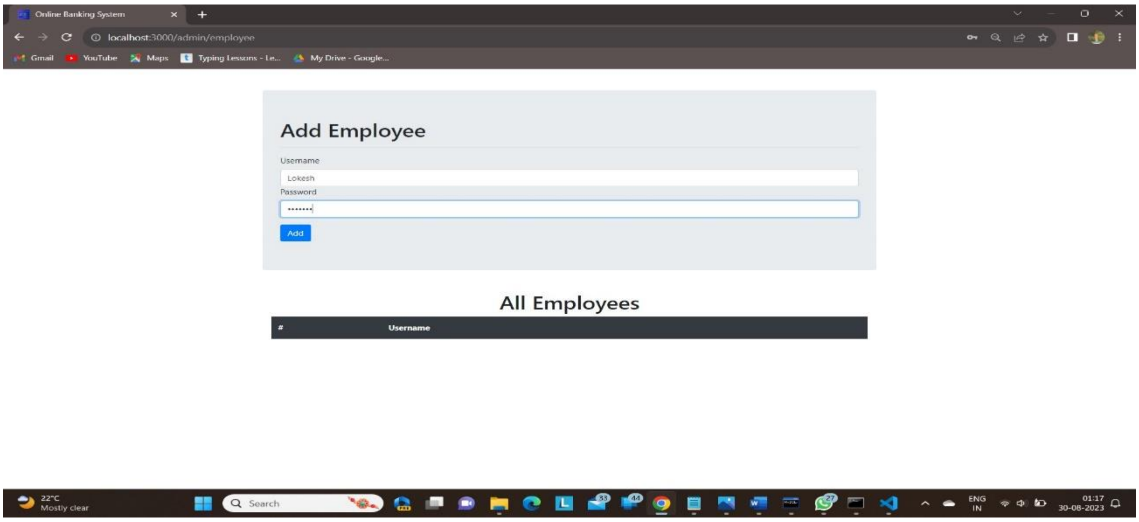
Zip Code: 425503

Username: mohit4235

Password: ••••••••••

Add

Adding an employee and employee list(Specific access) :



Add Employee

Username

Lokesh

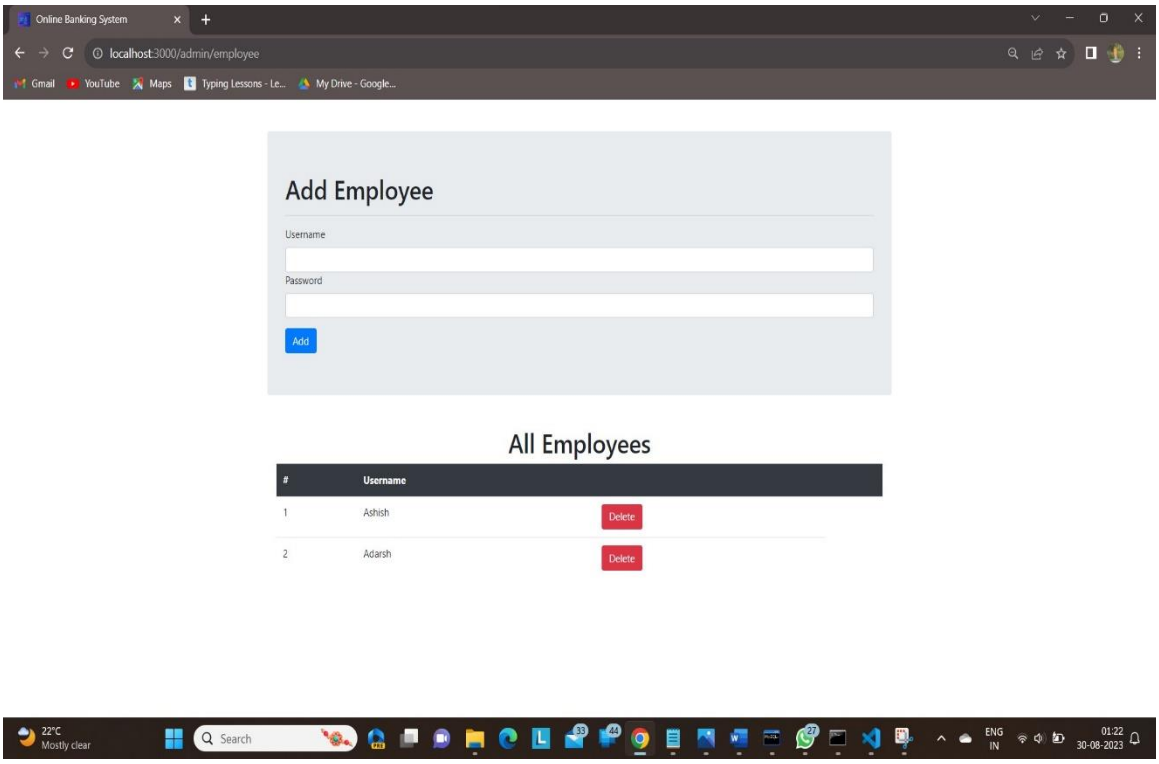
Password

•••••••

Add

All Employees

#	Username



Add Employee

Username

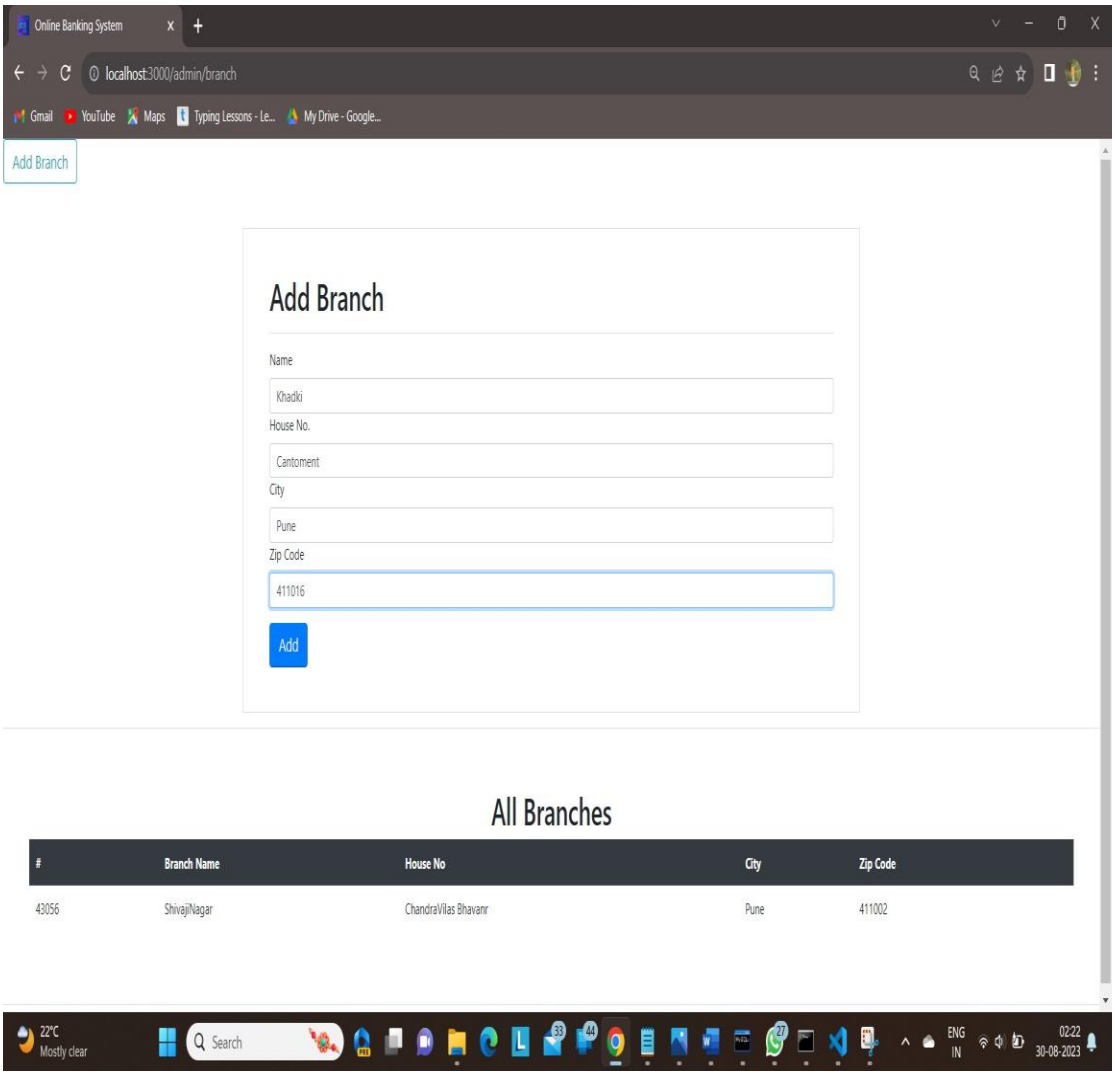
Password

Add

All Employees

#	Username	
1	Ashish	Delete
2	Adarsh	Delete

Add a branch and Branch list :



Add Branch

Name	Khadki
House No.	Cantoment
City	Pune
Zip Code	411016

All Branches

#	Branch Name	House No	City	Zip Code
43056	ShivajiNagar	ChandraVilas Bhavanr	Pune	411002

Future Scope:

- The Banking System Software, as outlined in the SRS, provides a solid foundation for a comprehensive banking solution. However, there are several avenues for future expansion and enhancement:

- **Online Banking and Mobile Apps**: Implement online banking features and develop mobile applications to enable customers to perform transactions and manage accounts remotely.

- **Advanced Analytics**: Enhance reporting and analytics capabilities to provide more insightful data for decision-making and trend analysis.

- **Multi-Currency Support**: Integrate support for multiple currencies to cater to international customers.

- **Intelligent Automation**: Incorporate artificial intelligence and machine learning for fraud detection, customer insights, and personalized services.

- **Integration with Payment Gateways**: Integrate with third-party payment gateways to facilitate online payments and fund transfers.

Conclusion:

In culmination, the Software Requirements Specification (SRS) for the Banking System Software encapsulates a meticulous blueprint for creating a robust and feature-rich banking solution. The comprehensive documentation has delineated the intricate functional and non-functional requirements across various modules, ensuring a holistic approach to managing banking operations.

The articulated SRS underscores the imperative aspects of security, performance, reliability, user experience, compliance, and analytics, all aimed at delivering an exemplary banking experience. The meticulous detailing of each module, from Employee and Customer Management to Account and Transaction Management, underscores the dedication to meeting the dynamic needs of the banking sector.

This SRS serves as a cornerstone for the development team, steering the course toward a system that aligns with industry standards and user expectations. Its meticulous delineation sets the trajectory for a solution capable of transforming banking operations, enhancing customer engagement, and fostering innovation. As this journey unfolds, the collaboration between developers, stakeholders, and end-users will pave the way for a sophisticated banking system that blends technological prowess with seamless functionality, ultimately shaping the future of modern banking.

References:

- https://www.w3schools.com/
- https://react-bootstrap.github.io/components/carousel/
- https://www.geeksforgeeks.org/reactjs-tutorials/
- https://javaee.github.io/javaee-spec/javadocs/
- https://reactjs.org/docs/getting-started.html