

CS306 Phase 2

PART 1: Trigger Implementation

1. Trigger: log_balance_update

Purpose:

This trigger automatically logs transactions whenever the balance of an account is updated.

Functionality:

- **Trigger Type:** AFTER UPDATE — The trigger executes after any update operation on the Account table.
- **Condition:**
 - Compares NEW.balance (the updated value) with OLD.balance (the value before the update).
 - If they are different, the system assumes the balance has changed and proceeds with the following action.
- **Action:**
 - Inserts a record into the Transaction table with:
 - account_number: The account that was updated.
 - transaction_type: Describes the action as "Balance Update."
 - amount: The new balance of the account.
 - transaction_date: The current date (CURDATE()).
 - description: A note indicating the balance was updated automatically.

This ensures that every balance update is recorded in the Transaction table for tracking purposes.

A SQL script for Trigger1:

```
-- Trigger1
DELIMITER //
CREATE TRIGGER log_balance_update
AFTER UPDATE ON Account
FOR EACH ROW
BEGIN
    -- Log only when the balance is updated
    IF NEW.balance <> OLD.balance THEN
        INSERT INTO Transaction (account_number, transaction_type, amount,
transaction_date, description)
VALUES (NEW.account_number, 'Balance Update', NEW.balance, CURDATE(),
'Balance updated automatically');
    END IF;
END;
//
```

DELIMITER ;

- **How do we observe this in MySQL?**

Before this trigger1, the account(a) and transaction(b) tables respectively.

	account_numbr...	customer_id	branch_id	balance	creation_da...	status
1	1	1	1	5000.00	2024-12-08	Active
2	2	2	2	3000.00	2024-12-08	Active
3	3	3	3	10000.00	2024-12-08	Active
4	4	4	4	7500.00	2024-12-08	Inactive
5	5	5	5	1500.00	2024-12-08	Active
	NULL	NULL	NULL	NULL	NULL	NULL

(a)

	transaction...	account_numbr...	transaction_ty...	amount	transaction_d...	description
1	1	1	Deposit	2000.00	2024-12-08	Initial deposit
2	2	2	Withdrawal	500.00	2024-12-08	ATM withdrawal
3	3	3	Deposit	3000.00	2024-12-08	Salary deposit
4	4	4	Transfer	1000.00	2024-12-08	Transfer to another account
5	5	5	Fee	50.00	2024-12-08	Account maintenance fee
	NULL	NULL	NULL	NULL	NULL	NULL

(b)

After running this update statement, the new tables will be c and d respectively:

UPDATE Account SET balance = 4500.00 WHERE account_number = 1;

	account_numbr...	customer_id	branch_id	balance	creation_da...	status
1	1	1	1	4500.00	2024-12-08	Active
2	2	2	2	3000.00	2024-12-08	Active
3	3	3	3	10000.00	2024-12-08	Active
4	4	4	4	7500.00	2024-12-08	Inactive
5	5	5	5	1500.00	2024-12-08	Active
	NULL	NULL	NULL	NULL	NULL	NULL

(c)

	transaction...	account_num...	transaction_ty...	amount	transaction_d...	description	
	1	1	Deposit	2000.00	2024-12-08	Initial deposit	
	2	2	Withdrawal	500.00	2024-12-08	ATM withdrawal	
	3	3	Deposit	3000.00	2024-12-08	Salary deposit	
	4	4	Transfer	1000.00	2024-12-08	Transfer to another account	
	5	5	Fee	50.00	2024-12-08	Account maintenance fee	
	6	1	Balance Update	4500.00	2024-12-08	Balance updated automatically	
	NULL	NULL	NULL	NULL	NULL	NULL	

(d)

2. Trigger: account_status_update

Purpose:

Logs a status change whenever the status of an account is updated.

Functionality:

- **Trigger Type:** AFTER UPDATE — Executes after any update operation on the Account table.
- **Condition:**
 - Compares NEW.status with OLD.status.
 - If they are different, the system assumes the status has changed.
- **Action:**
 - Inserts a record into the Transaction table with:
 - account_number: The account whose status was changed.
 - transaction_type: Describes the action as "Status Update."
 - amount: Logs 0.00 because status changes do not involve monetary transactions.
 - transaction_date: The current date (CURDATE()).
 - description: Includes a message detailing the change from the old status to the new status.

This ensures all account status changes (e.g., Active to Inactive) are tracked in the Transaction table.

A SQL script for Trigger2:

```
-- Trigger2 to check for the statuses - active/inactive
DELIMITER //
CREATE TRIGGER account_status_update
AFTER UPDATE ON Account
FOR EACH ROW
BEGIN
    -- Log the status update (only if the status has changed)
    IF NEW.status <> OLD.status THEN
        INSERT INTO Transaction (account_number, transaction_type, amount,
transaction_date, description)
VALUES (NEW.account_number, 'Status Update', 0.00, CURDATE(),
CONCAT('Account status changed from ', OLD.status, ' to ', NEW.status));
    END IF;
END //
```

```
END;  
//  
DELIMITER ;
```

- **How do we observe this in MySQL?**

The initial versions of tables are in figures a and b respectively. After running this update part, the new tables will be e and f respectively:

```
UPDATE Account  
SET status = 'Closed'  
WHERE account_number = 1;
```

	account_num...	customer_id	branch_id	balance	creation_da...	status	
	1	1	1	4500.00	2024-12-08	Closed	
	2	2	2	3000.00	2024-12-08	Active	
	3	3	3	10000.00	2024-12-08	Active	
	4	4	4	7500.00	2024-12-08	Inactive	
	5	5	5	1500.00	2024-12-08	Active	
	NULL	NULL	NULL	NULL	NULL	NULL	

(e)

	transaction...	account_num...	transaction_ty...	amount	transaction_d...	description	
	1	1	Deposit	2000.00	2024-12-08	Initial deposit	
	2	2	Withdrawal	500.00	2024-12-08	ATM withdrawal	
	3	3	Deposit	3000.00	2024-12-08	Salary deposit	
	4	4	Transfer	1000.00	2024-12-08	Transfer to another account	
	5	5	Fee	50.00	2024-12-08	Account maintenance fee	
	6	1	Balance Update	4500.00	2024-12-08	Balance updated automatically	
	7	1	Status Update	0.00	2024-12-08	Account status changed from Active to Closed	
	NULL	NULL	NULL	NULL	NULL	NULL	

(f)

PART 2: Stored Procedure Implementation

Purpose:

Handles the process of user login by verifying the user's role and credentials to ensure data accuracy and access control.

Functionality:

Input Parameters:

- role: Specifies the user's role, either "customer" or "employee".
- first_name: The first name of the user attempting to log in.
- last_name: The last name of the user attempting to log in.
- user_id: The unique identifier of the user, corresponding to their role (e.g., customer_id or employee_id).

Step-by-Step Actions:

1. Role Verification:
 - If the role is "customer", the procedure checks the Customer table for a record matching the provided first_name, last_name, and customer_id.
 - If the role is "employee", the procedure checks the Employee table for a record matching the provided first_name, last_name, and employee_id.
2. Error Handling:
 - If the role is neither "customer" nor "employee", the procedure raises an error using SIGNAL SQLSTATE with the message "Invalid role specified".
3. Result:
 - Returns the user ID (customer_id or employee_id) if a matching record is found. If no match is found, no result is returned.

Example Use Case:

The stored procedure is used when a user attempts to log in to the system. It ensures that the user is valid and authorized based on their role and credentials.

DELIMITER //

```
CREATE PROCEDURE UserLogin (
  IN role VARCHAR(20),
  IN first_name VARCHAR(50),
  IN last_name VARCHAR(50),
  IN user_id INT
)
BEGIN
  IF role = 'customer' THEN
    SELECT customer_id FROM Customer
    WHERE first_name = first_name
      AND last_name = last_name
      AND customer_id = user_id;
  ELSEIF role = 'employee' THEN
    SELECT employee_id FROM Employee
    WHERE first_name = first_name
      AND last_name = last_name
      AND employee_id = user_id;
  ELSE
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Invalid role specified';
  END IF;
END //
```

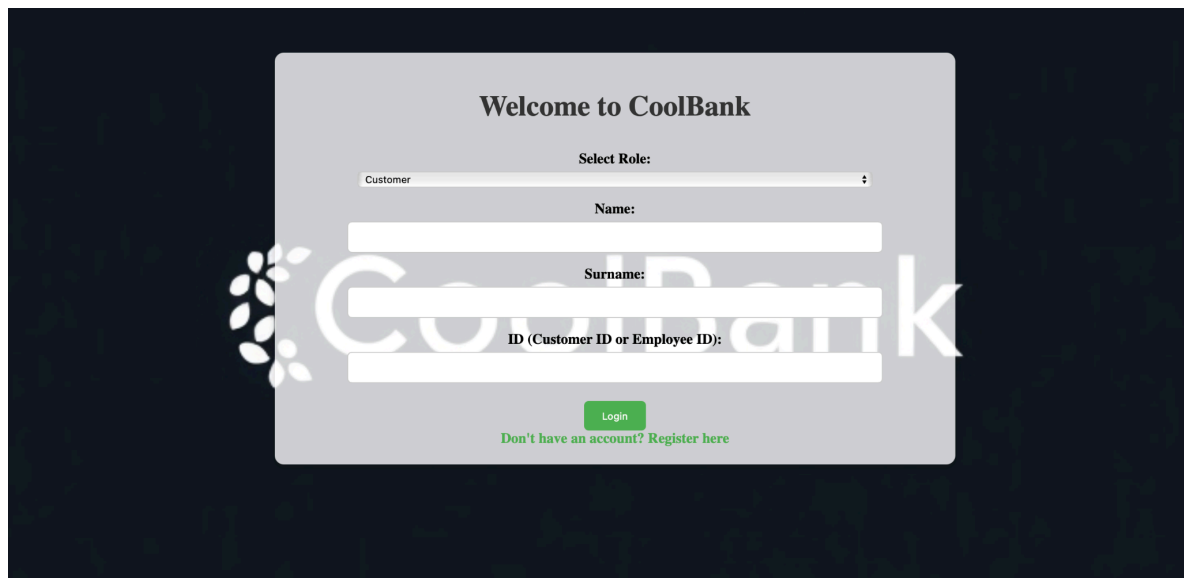
DELIMITER ;

The checking of the procedure will be done in part 3. Procedure is running in a PHP file, you can check for more details in Part3.

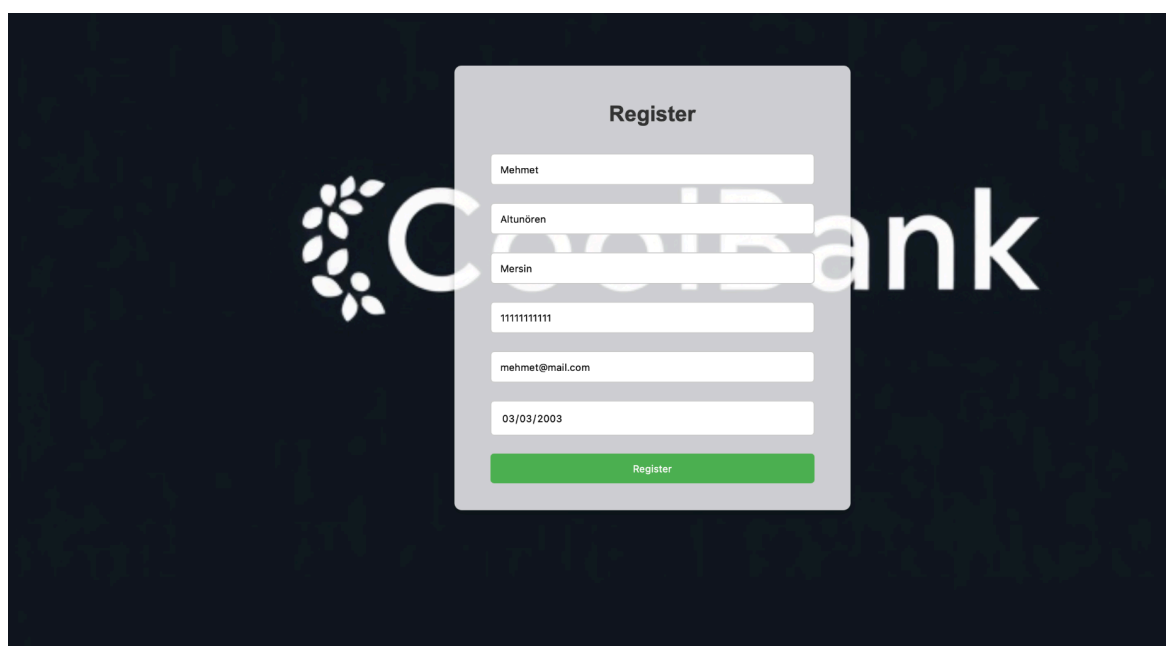
PART 3: Web Access Module

Screenshots of the following:

- **Input Form:** The web form where users can input data.

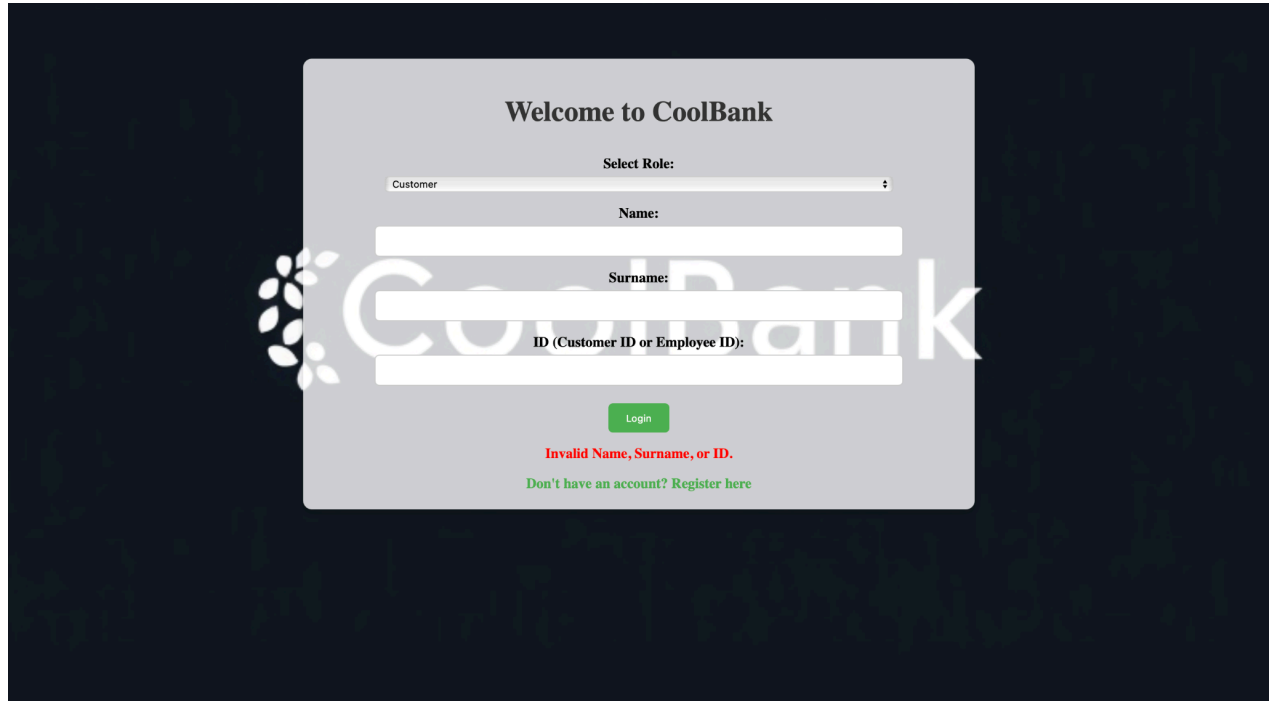
A screenshot of the CoolBank login interface. The page has a dark blue background with a large, faint 'CoolBank' logo. In the center, there is a light gray rectangular box containing the login form. The form is titled 'Welcome to CoolBank' in bold black text. Below the title, there is a 'Select Role:' dropdown menu with 'Customer' selected. Below this, there are three input fields labeled 'Name:', 'Surname:', and 'ID (Customer ID or Employee ID):'. At the bottom of the form, there is a green 'Login' button and a link that says 'Don't have an account? Register here' in green text.

On this page, there are input fields where customers or employees can enter data. If there is no account, there is a register option below. Only after creating an account can members access the login section and manage their account-related operations.

A screenshot of the CoolBank register interface. The page has a dark blue background with a large, faint 'CoolBank' logo. In the center, there is a light gray rectangular box containing the register form. The form is titled 'Register' in bold black text. Below the title, there are seven input fields: 'Name' (with 'Mehmet' entered), 'Surname' (with 'Altundren' entered), 'City' (with 'Mersin' entered), 'ID' (with '1111111111' entered), 'Email' (with 'mehmet@mail.com' entered), and 'Date of Birth' (with '03/03/2003' entered). At the bottom of the form, there is a green 'Register' button.

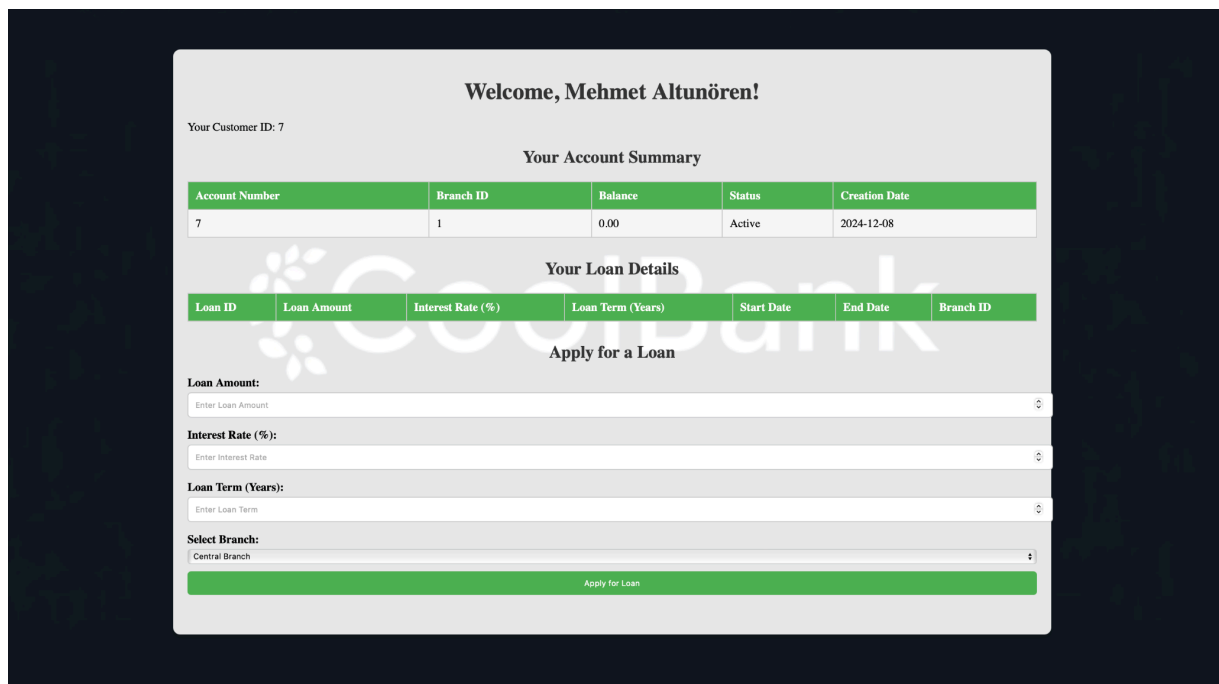
This page is the registration page, where users can create an account (insertion of new data into the system), become members of CoolBank, and manage their account-related operations.

- **Insertion Confirmation:** The output/results displayed on the web interface after interaction with the stored procedure.



The image shows a web interface for CoolBank. At the top, it says "Welcome to CoolBank". Below this is a "Select Role:" dropdown menu with "Customer" selected. There are three input fields for "Name:", "Surname:", and "ID (Customer ID or Employee ID):". A green "Login" button is below the ID field. A red error message "Invalid Name, Surname, or ID." is displayed below the button. At the bottom, there is a green link that says "Don't have an account? Register here".

As can be seen, when the inputs do not match, the UserLogin procedure works as expected and displays an error. But if it works correctly, below page is shown and customers can do account - related activities on this page.



The image shows a web interface for CoolBank, displaying a welcome message for a logged-in user. The header says "Welcome, Mehmet Altunören!". Below this, it says "Your Customer ID: 7". The main section is titled "Your Account Summary" and contains a table with the following data:

Account Number	Branch ID	Balance	Status	Creation Date
7	1	0.00	Active	2024-12-08

Below the table, there is a section titled "Your Loan Details" with a table that has the following columns: Loan ID, Loan Amount, Interest Rate (%), Loan Term (Years), Start Date, End Date, and Branch ID. Below this table, there is a section titled "Apply for a Loan" with the following form fields:

- Loan Amount: Enter Loan Amount
- Interest Rate (%): Enter Interest Rate
- Loan Term (Years): Enter Loan Term
- Select Branch: Central Branch

At the bottom of the form, there is a green button labeled "Apply for Loan".

- **Data View:** A display of the updated database content or results retrieved from the database.

Before any database insertion, there are predefined values inserted through the MySQL code shown in Figure 1. After the registration of new customers (deniz and mehmet), the newly inserted values (account for customers) can be observed in Figure 2.

Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08

(1)

Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	10,000.00	Active	2024-12-08
7	7	Mehmet Altunören	1	0.00	Active	2024-12-08

(2)

Also, employees can access their own dashboard, where employees can see the account summaries, loan details and customer details after these insertions.

Employee Dashboard

All Account Summaries

Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	10,000.00	Active	2024-12-08
7	7	Mehmet Altunören	1	0.00	Active	2024-12-08

All Customer Loan Details

Loan ID	Customer ID	Customer Name	Loan Amount	Interest Rate (%)	Loan Term (Years)	Start Date	End Date	Branch Name	Branch ID
1	1	Alice Smith	10,000.00	5.50	10	2024-12-08	2034-12-08	Central Branch	1
2	4	Daisy White	7,000.00	5.00	7	2024-12-08	2031-12-08	North Branch	4
2	2	Bob Jones	5,000.00	4.50	5	2024-12-08	2029-12-08	East Branch	2
5	5	Edward Green	8,000.00	4.80	8	2024-12-08	2032-12-08	South Branch	5
3	3	Charlie Brown	15,000.00	6.00	15	2024-12-08	2039-12-08	West Branch	3
6	7	Mehmet Altunören	1,000,000.00	1.00	10	2024-12-08	2034-12-08	West Branch	3

All Customer Details

Customer ID	First Name	Last Name	Address	Phone Number	Email	Date of Birth	Registration Date
1	Alice	Smith	123 Elm St	5551234567	alice.smith@example.com	1985-06-15	2024-12-08
2	Bob	Jones	456 Oak St	5557654321	bob.jones@example.com	1990-08-20	2024-12-08
3	Charlie	Brown	789 Pine St	5556781234	charlie.brown@example.com	1982-11-05	2024-12-08
4	Daisy	White	321 Maple St	5554327890	daisy.white@example.com	1995-04-12	2024-12-08
5	Edward	Green	654 Cedar St	5559876543	edward.green@example.com	1978-03-27	2024-12-08
6	Deniz	Kandemir	Mevsim	1111111111	deniz@mail.com	2003-04-24	2024-12-08
7	Mehmet	Altunören	Istanbul	0000000000	mehmet@mail.com	2000-01-01	2024-12-08

Update Account Details

Account Number:

Enter Account Number

New Balance (Optional):

Enter New Balance

Update Status (Optional):

Select Status

Update Account

Another insertion is loan insertion. Loan insertion can be done through the customer dashboard. A successful loan insertion (before and after versions) can be observed in Figure 3

and 4.

Welcome, Mehmet Altunören!

Your Customer ID: 7

Your Account Summary

Account Number	Branch ID	Balance	Status	Creation Date
----------------	-----------	---------	--------	---------------

Your Loan Details

Loan ID	Loan Amount	Interest Rate (%)	Loan Term (Years)	Start Date	End Date	Branch ID
---------	-------------	-------------------	-------------------	------------	----------	-----------

Apply for a Loan

Loan Amount:

1000000

Interest Rate (%):

0.1

Loan Term (Years):

10

Select Branch:

East Branch

Apply for Loan

All Customer Loan Details									
Loan ID	Customer ID	Customer Name	Loan Amount	Interest Rate (%)	Loan Term (Years)	Start Date	End Date	Branch Name	Branch ID
1	1	Alice Smith	10,000.00	5.50	10	2024-12-08	2034-12-08	Central Branch	1
2	2	Bob Jones	5,000.00	4.50	5	2024-12-08	2029-12-08	East Branch	2
3	3	Charlie Brown	15,000.00	6.00	15	2024-12-08	2039-12-08	West Branch	3
4	4	Daisy White	7,000.00	5.00	7	2024-12-08	2031-12-08	North Branch	4
5	5	Edward Green	8,000.00	4.80	8	2024-12-08	2032-12-08	South Branch	5

(3)

All Customer Loan Details									
Loan ID	Customer ID	Customer Name	Loan Amount	Interest Rate (%)	Loan Term (Years)	Start Date	End Date	Branch Name	Branch ID
1	1	Alice Smith	10,000.00	5.50	10	2024-12-08	2034-12-08	Central Branch	1
4	4	Daisy White	7,000.00	5.00	7	2024-12-08	2031-12-08	North Branch	4
2	2	Bob Jones	5,000.00	4.50	5	2024-12-08	2029-12-08	East Branch	2
5	5	Edward Green	8,000.00	4.80	8	2024-12-08	2032-12-08	South Branch	5
3	3	Charlie Brown	15,000.00	6.00	15	2024-12-08	2039-12-08	West Branch	3
6	7	Mehmet Altunören	1,000,000.00	1.00	10	2024-12-08	2034-12-08	West Branch	3

(4)

How do our triggers work?

Account is updated by employees only. After they input new account information, in here status, update is done in the database. If there is any change in the account balance, the change is made via a trigger and updates automatically. The account balances before and after the change can be observed in Figures 5 and 6, respectively.

All Account Summaries						
Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	10,000.00	Active	2024-12-08
7	7	Mehmet Altunören	1	0.00	Active	2024-12-08

(5)

Update Account Details

Account Number:

7

New Balance (Optional):

1000000

Update Status (Optional):

Active

Update Account

(the change is done here)

All Account Summaries						
Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	10,000.00	Active	2024-12-08
7	7	Mehmet Altunören	1	1,000,000.00	Active	2024-12-08

(6)

If there is any change in the account status, the change is made via a trigger and updates automatically. The account status before and after the change can be observed in Figures 7 and 8, respectively.

All Account Summaries						
Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	10,000.00	Active	2024-12-08
7	7	Mehmet Altunören	1	1,000,000.00	Active	2024-12-08

(7)

Update Account Details

Account Number:

6

New Balance (Optional):

Enter New Balance

Update Status (Optional):

Closed

Update Account

(the change is done here)

All Account Summaries						
Account Number	Customer ID	Customer Name	Branch ID	Balance	Status	Creation Date
1	1	Alice Smith	1	4,500.00	Closed	2024-12-08
2	2	Bob Jones	2	3,000.00	Active	2024-12-08
3	3	Charlie Brown	3	10,000.00	Active	2024-12-08
4	4	Daisy White	4	7,500.00	Inactive	2024-12-08
5	5	Edward Green	5	1,500.00	Active	2024-12-08
6	6	Deniz Kandemir	1	0.00	Closed	2024-12-08
7	7	Mehmet Altunören	1	1,000,000.00	Active	2024-12-08

(8)

- **HTML and backend files(.php) for the web interface:**

register.php:

```
<?php
session_start();

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "CoolBank";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $first_name = $_POST['first_name'];
    $last_name = $_POST['last_name'];
    $address = $_POST['address'];
    $phone_number = $_POST['phone_number'];
    $email = $_POST['email'];
    $date_of_birth = $_POST['date_of_birth'];
    $registration_date = date('Y-m-d'); // Current date

    // Generate a new customer ID (incrementing the max current customer_id)
    $id_query = "SELECT MAX(customer_id) AS max_id FROM Customer";
    $id_result = $conn->query($id_query);
    $row = $id_result->fetch_assoc();
    $new_customer_id = $row['max_id'] + 1;

    // Insert new customer into the Customer table
```

```

    $insert_sql = "INSERT INTO Customer (customer_id, first_name, last_name,
address, phone_number, email, date_of_birth, registration_date)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    $stmt = $conn->prepare($insert_sql);
    $stmt->bind_param("isssssss", $new_customer_id, $first_name, $last_name,
$address, $phone_number, $email, $date_of_birth, $registration_date);

    if ($stmt->execute()) {
        // Registration successful, set session and redirect to dashboard
        $_SESSION['customer_id'] = $new_customer_id;
        header("Location: dashboard.php");
        exit();
    } else {
        $error = "Error: " . $stmt->error;
    }

    $stmt->close();
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
    <style>
        body {
            background-image: url('background.jpg');
            background-size: cover;
            font-family: Arial, sans-serif;
            color: #333;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .container {
            background-color: rgba(255, 255, 255, 0.8);
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            width: 400px;

```

```

        text-align: center;
    }
    h2 {
        margin-bottom: 20px;
    }
    form input, form button {
        width: 90%;
        padding: 10px;
        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
    }
    form button {
        background-color: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
    }
    form button:hover {
        background-color: #45a049;
    }
    .error {
        color: red;
        font-weight: bold;
        margin-top: 10px;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Register</h2>
        <form action="" method="post">
            <input type="text" name="first_name" placeholder="First Name"
required>
            <input type="text" name="last_name" placeholder="Last Name"
required>
            <input type="text" name="address" placeholder="Address" required>
            <input type="text" name="phone_number" placeholder="Phone Number"
required>
            <input type="email" name="email" placeholder="Email" required>
            <input type="date" name="date_of_birth" required>
            <button type="submit">Register</button>
        </form>
        <!-- Display error message if registration fails -->
        <?php if (isset($error)) { ?>

```

```

        <p class="error"><?php echo $error; ?></p>
    <?php } ?>
</div>
</body>
</html>

```

login.php:

```

<?php
session_start();

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "CoolBank";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Process the login form if submitted
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $role = $_POST['role'];
    $name = $_POST['name'];
    $surname = $_POST['surname'];
    $id = $_POST['id'];

    // Login logic for customer
    if ($role === 'customer') {
        $sql = "SELECT * FROM Customer WHERE first_name = ? AND last_name = ?
AND customer_id = ?";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("ssi", $name, $surname, $id);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
            // Login successful
            $customer = $result->fetch_assoc();
            $_SESSION['customer_id'] = $customer['customer_id'];
            header("Location: dashboard.php");

```

```

        exit();
    } else {
        // Invalid credentials
        $error = "Invalid Name, Surname, or ID.";
    }
}

// Login logic for employee
if ($role === 'employee') {
    $sql = "SELECT * FROM Employee WHERE first_name = ? AND last_name = ?
AND employee_id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ssi", $name, $surname, $id);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // Login successful
        $_SESSION['employee_id'] = $id;
        header("Location: employee_dashboard.php");
        exit();
    } else {
        // Invalid credentials
        $error = "Invalid Name, Surname, or ID.";
    }
}

$stmt->close();
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
    <style>
        body {
            font-family: 'Garamond', serif;
            background: url('background.jpg') no-repeat center center fixed;
            background-size: cover;
            margin: 0;
            padding: 0;

```

```
}

.container {
  width: 50%;
  margin: 5% auto;
  padding: 20px;
  background-color: rgba(255, 255, 255, 0.8);
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  text-align: center;
}

h1 {
  color: #333;
}

form {
  display: flex;
  flex-direction: column;
  align-items: center;
}

label {
  margin-top: 10px;
  font-weight: bold;
}

input, select {
  width: 80%;
  padding: 10px;
  margin: 5px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
}

button {
  margin-top: 15px;
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
```



```

        background-color: #45a049;
    }

    .error {
        color: red;
        font-weight: bold;
        margin-top: 15px;
    }

    .register-link {
        margin-top: 20px;
        text-decoration: none;
        color: #4CAF50;
        font-weight: bold;
    }

    .register-link:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Welcome to CoolBank</h1>
        <form action="login.php" method="POST">
            <label for="role">Select Role:</label>
            <select id="role" name="role" required>
                <option value="customer">Customer</option>
                <option value="employee">Employee</option>
            </select>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>
            <label for="surname">Surname:</label>
            <input type="text" id="surname" name="surname" required>
            <label for="id">ID (Customer ID or Employee ID):</label>
            <input type="password" id="id" name="id" required>
            <button type="submit">Login</button>
        </form>
        <!-- Display error message if invalid login -->
        <?php if (isset($error)) { ?>
            <p class="error"><?php echo $error; ?></p>
        <?php } ?>
        <a href="register.php" class="register-link">Don't have an account?
Register here</a>
    </div>
</body>

```

</html>

dashboard.php

```
<?php
session_start();

// Check if the user is logged in
if (!isset($_SESSION['customer_id'])) {
    header("Location: login.php?error=1");
    exit();
}

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$database = "CoolBank";

$conn = new mysqli($servername, $username, $password, $database);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch customer details
$customer_id = $_SESSION['customer_id'];
$sql = "SELECT * FROM Customer WHERE customer_id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $customer_id);
$stmt->execute();
$result = $stmt->get_result();
$customer = $result->fetch_assoc();
$stmt->close();

// Fetch account summary
$accounts_sql = "SELECT * FROM Account WHERE customer_id = ?";
$stmt = $conn->prepare($accounts_sql);
$stmt->bind_param("i", $customer_id);
$stmt->execute();
$accounts = $stmt->get_result();
$stmt->close();

// Fetch loans for the logged-in customer
$loans_sql = "SELECT * FROM Loan WHERE customer_id = ?";
$stmt = $conn->prepare($loans_sql);
$stmt->bind_param("i", $customer_id);
```

```

$stmt->execute();
$loans = $stmt->get_result();
$stmt->close();

// Fetch branches for dropdown
$branches_sql = "SELECT branch_id, branch_name FROM Branch";
$branches_result = $conn->query($branches_sql);
$branches = [];
while ($row = $branches_result->fetch_assoc()) {
    $branches[] = $row;
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Customer Dashboard</title>
    <style>
        body {
            font-family: 'Garamond', serif;
            background: url('background.jpg') no-repeat center center fixed;
            background-size: cover;
            margin: 0;
            padding: 0;
        }

        .container {
            width: 70%;
            margin: 5% auto;
            padding: 20px;
            background-color: rgba(255, 255, 255, 0.9);
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
        }

        h1, h2 {
            color: #333;
            text-align: center;
        }

        .section {
            margin-bottom: 30px;

```

```
}

table {
    border-collapse: collapse;
    width: 100%;
    margin: 20px 0;
}

th, td {
    padding: 10px;
    text-align: left;
    border: 1px solid #ccc;
}

th {
    background-color: #4CAF50;
    color: white;
}

td {
    background-color: #f9f9f9;
}

label {
    font-weight: bold;
    display: block;
    margin-top: 10px;
}

input, select, button {
    width: 100%;
    padding: 10px;
    margin: 5px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
```

```

    }

    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome, <?php echo htmlspecialchars($customer['first_name'] . ' ' . $customer['last_name']); ?>!</h1>
        <p>Your Customer ID: <?php echo htmlspecialchars($customer['customer_id']); ?></p>

        <div class="section">
            <h2>Your Account Summary</h2>
            <table>
                <thead>
                    <tr>
                        <th>Account Number</th>
                        <th>Branch ID</th>
                        <th>Balance</th>
                        <th>Status</th>
                        <th>Creation Date</th>
                    </tr>
                </thead>
                <tbody>
                    <?php while ($account = $accounts->fetch_assoc()): ?>
                        <tr>
                            <td><?php echo htmlspecialchars($account['account_number']); ?></td>
                            <td><?php echo htmlspecialchars($account['branch_id']); ?></td>
                            <td><?php echo number_format($account['balance'], 2); ?></td>
                            <td><?php echo htmlspecialchars($account['status']); ?></td>
                            <td><?php echo htmlspecialchars($account['creation_date']); ?></td>
                        </tr>
                    <?php endwhile; ?>
                </tbody>
            </table>
        </div>

        <div class="section">
            <h2>Your Loan Details</h2>
            <table>
                <thead>
                    <tr>

```

```

        <th>Loan ID</th>
        <th>Loan Amount</th>
        <th>Interest Rate (%)</th>
        <th>Loan Term (Years)</th>
        <th>Start Date</th>
        <th>End Date</th>
        <th>Branch ID</th>
    </tr>
</thead>
<tbody>
    <?php while ($loan = $loans->fetch_assoc()): ?>
        <tr>
            <td><?php echo htmlspecialchars($loan['loan_id']);
?></td>

            <td><?php echo number_format($loan['loan_amount'],
2); ?></td>

            <td><?php echo
htmlspecialchars($loan['interest_rate']); ?></td>
            <td><?php echo
htmlspecialchars($loan['loan_term']); ?></td>
            <td><?php echo
htmlspecialchars($loan['start_date']); ?></td>
            <td><?php echo
htmlspecialchars($loan['end_date']); ?></td>
            <td><?php echo
htmlspecialchars($loan['branch_id']); ?></td>
        </tr>
    <?php endwhile; ?>
</tbody>
</table>
</div>

<div class="section">
    <h2>Apply for a Loan</h2>
    <form action="apply_loan.php" method="POST">
        <label for="loan_amount">Loan Amount:</label>
        <input type="number" id="loan_amount" name="loan_amount"
step="0.01" placeholder="Enter Loan Amount" required>

        <label for="interest_rate">Interest Rate (%):</label>
        <input type="number" id="interest_rate" name="interest_rate"
step="0.01" placeholder="Enter Interest Rate" required>

        <label for="loan_term">Loan Term (Years):</label>
        <input type="number" id="loan_term" name="loan_term"
placeholder="Enter Loan Term" required>

```

```

        <label for="branch_id">Select Branch:</label>
        <select id="branch_id" name="branch_id" required>
            <?php foreach ($branches as $branch): ?>
                <option value="<?php echo
htmlspecialchars($branch['branch_id']); ?>">
                    <?php echo
htmlspecialchars($branch['branch_name']); ?>
                </option>
            <?php endforeach; ?>
        </select>

        <button type="submit">Apply for Loan</button>
    </form>
</div>
</div>
</body>
</html>

```

employee_dashboard.php:

```

<?php
session_start();

// Check if the user is logged in as an employee
if (!isset($_SESSION['employee_id'])) {
    header("Location: login.php?error=1");
    exit();
}

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "CoolBank";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch all accounts and their associated customer details
$sql = "SELECT Account.account_number, Customer.customer_id,
Customer.first_name, Customer.last_name,

```

```

        Account.branch_id, Account.balance, Account.status,
Account.creation_date
        FROM Account
        JOIN Customer ON Account.customer_id = Customer.customer_id
        ORDER BY Account.creation_date DESC";
$result = $conn->query($sql);

// Fetch all loans and associated customer and branch details
$loans_sql = "SELECT Loan.loan_id, Loan.customer_id, Customer.first_name AS
customer_first_name,
        Customer.last_name AS customer_last_name,
Loan.loan_amount, Loan.interest_rate,
        Loan.loan_term, Loan.start_date, Loan.end_date,
Loan.branch_id, Branch.branch_name
        FROM Loan
        JOIN Customer ON Loan.customer_id = Customer.customer_id
        JOIN Branch ON Loan.branch_id = Branch.branch_id
        ORDER BY Loan.start_date DESC";
$loans = $conn->query($loans_sql);

// Fetch all customers
$customers_sql = "SELECT * FROM Customer ORDER BY registration_date DESC";
$customers = $conn->query($customers_sql);

// Handle account update
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['update_account']))
{
    $account_number = $_POST['account_number'];
    $new_balance = $_POST['balance'];
    $new_status = $_POST['status'];

    // Update account balance and/or status
    $update_sql = "UPDATE Account SET balance = IFNULL(?, balance), status =
IFNULL(?, status) WHERE account_number = ?";
    $stmt = $conn->prepare($update_sql);
    $stmt->bind_param("dsi", $new_balance, $new_status, $account_number);

    if ($stmt->execute()) {
        $update_message = "Account updated successfully for Account Number
$account_number.";
    } else {
        $update_error = "Failed to update account: " . $conn->error;
    }

    $stmt->close();
}

```



```
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Employee Dashboard</title>
  <style>
    body {
      font-family: 'Garamond', serif;
      background: url('background.jpg') no-repeat center center fixed;
      background-size: cover;
      margin: 0;
      padding: 0;
    }

    .container {
      width: 80%;
      margin: 5% auto;
      padding: 20px;
      background-color: rgba(255, 255, 255, 0.9);
      border-radius: 10px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
    }

    h1, h2 {
      color: #333;
      text-align: center;
    }

    .section {
      margin-bottom: 30px;
    }

    table {
      border-collapse: collapse;
      width: 100%;
      margin: 20px 0;
    }

    th, td {
      padding: 10px;
      text-align: left;
```

```
        border: 1px solid #ccc;
    }

    th {
        background-color: #4CAF50;
        color: white;
    }

    td {
        background-color: #f9f9f9;
    }

    form {
        margin-top: 20px;
    }

    input, select, button {
        width: 100%;
        padding: 10px;
        margin: 5px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    button {
        background-color: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
    }

    button:hover {
        background-color: #45a049;
    }

    .message {
        color: green;
        font-weight: bold;
        text-align: center;
    }

    .error {
        color: red;
        font-weight: bold;
        text-align: center;
    }
}
```

```

</style>
</head>
<body>
    <div class="container">
        <h1>Employee Dashboard</h1>

        <!-- Display success or error messages -->
        <?php if (isset($update_message)): ?>
            <p class="message"><?php echo $update_message; ?></p>
        <?php endif; ?>
        <?php if (isset($update_error)): ?>
            <p class="error"><?php echo $update_error; ?></p>
        <?php endif; ?>

        <!-- Accounts Summary -->
        <div class="section">
            <h2>All Account Summaries</h2>
            <table>
                <thead>
                    <tr>
                        <th>Account Number</th>
                        <th>Customer ID</th>
                        <th>Customer Name</th>
                        <th>Branch ID</th>
                        <th>Balance</th>
                        <th>Status</th>
                        <th>Creation Date</th>
                    </tr>
                </thead>
                <tbody>
                    <?php while ($row = $result->fetch_assoc()): ?>
                        <tr>
                            <td><?php echo $row['account_number']; ?></td>
                            <td><?php echo $row['customer_id']; ?></td>
                            <td><?php echo $row['first_name'] . ' ' .
$row['last_name']; ?></td>
                            <td><?php echo $row['branch_id']; ?></td>
                            <td><?php echo number_format($row['balance'], 2);
?></td>
                            <td><?php echo $row['status']; ?></td>
                            <td><?php echo $row['creation_date']; ?></td>
                        </tr>
                    <?php endwhile; ?>
                </tbody>
            </table>
        </div>
    </div>

```

```

<!-- Loans Summary -->
<div class="section">
    <h2>All Customer Loan Details</h2>
    <table>
        <thead>
            <tr>
                <th>Loan ID</th>
                <th>Customer ID</th>
                <th>Customer Name</th>
                <th>Loan Amount</th>
                <th>Interest Rate (%)</th>
                <th>Loan Term (Years)</th>
                <th>Start Date</th>
                <th>End Date</th>
                <th>Branch Name</th>
                <th>Branch ID</th>
            </tr>
        </thead>
        <tbody>
            <?php while ($loan = $loans->fetch_assoc()): ?>
                <tr>
                    <td><?php echo $loan['loan_id']; ?></td>
                    <td><?php echo $loan['customer_id']; ?></td>
                    <td><?php echo $loan['customer_first_name'] . ' ' . $loan['customer_last_name']; ?></td>
                    <td><?php echo number_format($loan['loan_amount'],
2); ?></td>
                    <td><?php echo $loan['interest_rate']; ?></td>
                    <td><?php echo $loan['loan_term']; ?></td>
                    <td><?php echo $loan['start_date']; ?></td>
                    <td><?php echo $loan['end_date']; ?></td>
                    <td><?php echo $loan['branch_name']; ?></td>
                    <td><?php echo $loan['branch_id']; ?></td>
                </tr>
            <?php endwhile; ?>
        </tbody>
    </table>
</div>

<!-- Customers Summary -->
<div class="section">
    <h2>All Customer Details</h2>
    <table>
        <thead>
            <tr>

```

```

        <th>Customer ID</th>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Address</th>
        <th>Phone Number</th>
        <th>Email</th>
        <th>Date of Birth</th>
        <th>Registration Date</th>
    </tr>
</thead>
<tbody>
    <?php while ($customer = $customers->fetch_assoc()): ?>
        <tr>
            <td><?php echo $customer['customer_id']; ?></td>
            <td><?php echo $customer['first_name']; ?></td>
            <td><?php echo $customer['last_name']; ?></td>
            <td><?php echo $customer['address']; ?></td>
            <td><?php echo $customer['phone_number']; ?></td>
            <td><?php echo $customer['email']; ?></td>
            <td><?php echo $customer['date_of_birth']; ?></td>
            <td><?php echo $customer['registration_date'];
?></td>

        </tr>
    <?php endwhile; ?>
</tbody>
</table>
</div>

<!-- Update Account Form -->
<div class="section">
    <h2>Update Account Details</h2>
    <form action="employee_dashboard.php" method="POST">
        <label for="account_number">Account Number:</label>
        <input type="number" id="account_number" name="account_number"
placeholder="Enter Account Number" required>

        <label for="balance">New Balance (Optional):</label>
        <input type="number" id="balance" name="balance" step="0.01"
placeholder="Enter New Balance">

        <label for="status">Update Status (Optional):</label>
        <select id="status" name="status">
            <option value="">Select Status</option>
            <option value="Active">Active</option>
            <option value="Inactive">Inactive</option>
            <option value="Closed">Closed</option>

```

```

        </select>

        <button type="submit" name="update_account">Update
Account</button>

    </form>

</div>

</div>
</body>
</html>

```

apply_loan.php:

```

<?php
session_start();

// Check if the user is logged in
if (!isset($_SESSION['customer_id'])) {
    header("Location: login.php?error=1");
    exit();
}

// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "CoolBank";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $loan_amount = $_POST['loan_amount'];
    $interest_rate = $_POST['interest_rate'];
    $loan_term = $_POST['loan_term'];
    $branch_id = $_POST['branch_id'];
    $customer_id = $_SESSION['customer_id'];

    // Validate inputs
    if ($loan_amount <= 0 || $interest_rate <= 0 || $loan_term <= 0) {
        $error_message = "All fields must have positive values.";
    } else {

```

```

        // Call the ReserveLoan stored procedure
        $sql = $conn->prepare("CALL ReserveLoan(?, ?, ?, ?, ?)");
        $sql->bind_param("idddi", $customer_id, $loan_amount, $interest_rate,
$loan_term, $branch_id);

        if ($sql->execute()) {
            $success_message = "Loan successfully applied!";
        } else {
            $error_message = "Error: " . $conn->error;
        }
        $sql->close();
    }
}

// Fetch branches for dropdown
$branches_query = "SELECT branch_id, branch_name FROM Branch";
$branches_result = $conn->query($branches_query);
$branches = [];
while ($row = $branches_result->fetch_assoc()) {
    $branches[] = $row;
}

$conn->close();
?>

```

update_account_employee.php:

```

<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $customer_id = $_POST['customer_id'];
    $balance = $_POST['balance'];
    $status = $_POST['status'];

    $servername = "localhost";
    $username = "root";
    $password = "";
    $database = "CoolBank";

    $conn = new mysqli($servername, $username, $password, $database);
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Update account logic
    $sql = "UPDATE Account SET ";

```

```
$updates = [];  
if (!empty($balance)) {  
    $updates[] = "balance = $balance";  
}  
if (!empty($status)) {  
    $updates[] = "status = '$status'";  
}  
$sql .= implode(", ", $updates);  
$sql .= " WHERE customer_id = $customer_id";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Account updated successfully!";  
} else {  
    echo "Error: " . $conn->error;  
}  
  
$conn->close();  
}  
?>
```