Banking Information System - Project Report

1. Introduction

The Banking Information System is a web-based application designed to provide users with secure and efficient online banking functionalities. The system enables users to perform essential banking operations such as deposits, withdrawals, fund transfers, and account statement generation. Developed using Java Servlets and JSP, the application follows a structured approach to ensure smooth transaction processing and data management.

The project emphasizes security by implementing authentication mechanisms, password hashing, and session management. It also ensures data consistency through transaction management techniques in MySQL.

2. Project Objectives

- Develop a secure and reliable banking application.
- Enable seamless banking operations for users.
- Implement authentication and password security using hashing techniques.
- Maintain detailed transaction logs for auditing and tracking purposes.
- Ensure data integrity through proper database handling and validation.

3. Technology Stack

- Frontend: JSP (JavaServer Pages), HTML, CSS
- Backend: Java Servlets, JDBC (Java Database Connectivity)
- Database: MySQL
- Development Environment: Eclipse IDE
- Server: Apache Tomcat
- Security: Password hashing using SHA-256 with salt, HTTPS for secure communication

4. System Architecture

- The Banking Information System follows a three-tier architecture:
- Presentation Layer: JSP pages for user interaction and form submissions.
- Business Logic Layer: Java Servlets to process user requests and interact with the database.
- Data Layer: MySQL database for secure data storage and retrieval.
- This architecture ensures separation of concerns, scalability, and maintainability.

5. Modules and Features

5.1 User Management

- User Registration: New users can create an account by providing personal details. Passwords are hashed before storing in the database.
- User Login & Authentication: Secure login mechanism using password hashing and validation against the database. □ Logout Feature: Proper session invalidation to prevent unauthorized access after logout.

5.2 Transaction Management

- *DepositServlet:* Enables users to deposit money into their accounts, updating balance accordingly.
- *WithdrawServlet:* Allows users to withdraw money with validation to prevent overdrafts.
- *TransferServlet:* Facilitates fund transfers between accounts while maintaining transaction logs and validating available balance.

5.3 Account Management

- *TransactionServlet:* Retrieves and displays transaction history from the database.
- Account Details View: Displays account balance and recent transactions for user transparency.

6. Implementation Details

- The system is implemented using Java Servlets and JSP, where each module corresponds to a specific servlet that interacts with the database via JDBC.
- *LoginServlet:* Authenticates users by verifying hashed passwords stored in MySQL.
- *RegisterServlet:* Hashes user passwords using SHA-256 with salt before storing in the database.
- *TransactionServlet:* Fetches and displays transaction history with filtering options.
- *LogoutServlet:* Invalidates the session and prevents unauthorized re-access.
- Security Enhancements
- *Password Hashing:* Implemented SHA-256 hashing with salt for secure password storage.
- *Session Management:* Implemented session timeout and HTTPS to enhance security. *SQL Injection Prevention:* Used Prepared Statements to prevent SQL injection attacks.

7. Database Design

- The system utilizes the following database tables:
- Users: Stores user credentials, hashed passwords, and personal details.
- Accounts: Manages account balances and links users to their respective accounts.
- Transactions: Logs all deposit, withdrawal, and transfer transactions with timestamps.

Database Schema (Simplified)

```
CREATE
TABLE Users (
   user_id INT PRIMARY KEY AUTO_INCREMENT,
username VARCHAR(50) UNIQUE,     password_hash
VARCHAR(256),     salt VARCHAR(50),
   email VARCHAR(100),
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE Accounts (
   account_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT,
   balance DECIMAL(10,2) DEFAULT 0.00,
   FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

CREATE TABLE Transactions (
   transaction_id INT PRIMARY KEY AUTO_INCREMENT,
account_id INT,
   type ENUM('deposit', 'withdrawal', 'transfer'),
   amount DECIMAL(10,2),
   transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
);
```

8. Challenges and Solutions

8.1 Session Management

- Issue: Session hijacking or unauthorized access after logout.
- Solution: Implemented session timeout and HTTPS encryption.

8.2 Concurrency Issues

- Issue: Concurrent transactions could lead to incorrect balance updates.
- Solution: Used SQL transaction management with BEGIN TRANSACTION, COMMIT, and ROLLBACK to maintain data integrity.

8.3 Secure Authentication

- Issue: Storing plain text passwords poses security risks.
- Solution: Implemented SHA-256 hashing with salt to secure passwords.

8.4 Input Validation

- Issue: Improper user input could lead to SQL injection or broken functionality.
- Solution: Applied both client-side (JSP validation) and server-side (Servlets) input validation.

9. Future Enhancements

- Two-Factor Authentication: Adding OTP-based verification for enhanced security.
- Loan and Credit Card Management: Introducing features for users to apply for loans and manage credit cards.
- Enhanced UI: Modernizing the front-end using frameworks like React or Bootstrap. Mobile App Integration: Developing a mobile-friendly interface for easier access.

10.JAVA FILES:

```java
package dao;

import java.sql.Connection;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/BankingManagement";
    private static final String USER = "root";
    private static final String PASSWORD = "Sprathap#30";

    public static Connection getConnection() {
        Connection conn = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return conn;
    }
}
```

Login Servlet:

```java
package servlet;

import java.io.IOException;

@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        String emailOrContact = request.getParameter("emailOrContact");
        String password = request.getParameter("password");

        try (Connection con = DatabaseConnection.getConnection())
        {
            String query = "SELECT user_id, name, account_number, password FROM users WHERE email = ? OR contact = ?";
            try (PreparedStatement ps = con.prepareStatement(query))
            {
                ps.setString(1, emailOrContact);
                ps.setString(2, emailOrContact);

                try (ResultSet rs = ps.executeQuery())
                {
                    if (rs.next())
                    {
                        String hashedPassword = rs.getString("password");

                        if (BCrypt.checkpw(password, hashedPassword))
                        {
                            HttpSession session = request.getSession();
                            session.setAttribute("user_id", rs.getInt("user_id"));
                            session.setAttribute("name", rs.getString("name"));
                            session.setAttribute("account_number", rs.getString("account_number"));

                            response.sendRedirect("dashboard.jsp"); // Redirect to dashboard after login
                        } else
                        {
                            response.getWriter().println("Invalid credentials. Try again.");
                        }
                    } else
                    {
                        response.getWriter().println("Invalid credentials. Try again.");
                    }
                }
            }
        } catch (Exception e)
        {
            e.printStackTrace();
            response.getWriter().println("Error: " + e.getMessage());
        }
    }
}
```

Logout servlet:

```java
package servlet;

import java.io.IOException;

@WebServlet("/LogoutServlet")
public class LogoutServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        session.invalidate(); // Destroy session
        response.sendRedirect("login.jsp"); // Redirect to login page
    }
}
```

Register Servlet:

```java
package servlet;

import java.io.IOException;

@WebServlet("/RegisterServlet")
public class RegisterServlet extends HttpServlet
{
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {
        String name = request.getParameter("name");
        String address = request.getParameter("address");
        String contact = request.getParameter("contact");
        String email = request.getParameter("email");
        String password = request.getParameter("password");

        // Hash password before storing it in DB
        String hashedPassword = BCrypt.hashpw(password, BCrypt.gensalt());

        try (Connection con = DatabaseConnection.getConnection())
        {
            // Start transaction
            con.setAutoCommit(false);

            // Generate Unique Account Number
            long accountNumber = generateUniqueAccountNumber(con);

            // Insert into users table
            String query = "INSERT INTO users (name, address, contact, email, password, account_number) VALUES (?, ?, ?, ?, ?, ?)";
            try (PreparedStatement ps = con.prepareStatement(query))
            {
                ps.setString(1, name);
                ps.setString(2, address);
                ps.setString(3, contact);
                ps.setString(4, email);
                ps.setString(5, hashedPassword);
                ps.setLong(6, accountNumber);

                int rowsInserted = ps.executeUpdate();
                if (rowsInserted > 0)
                {
                    // Insert account entry in accounts table
```

```java
                    // Insert account entry in accounts table
                    String accountQuery = "INSERT INTO accounts (account_number, user_id, balance) VALUES (?, (SELECT user_id FROM users WHERE email = ?), ?)";
                    try (PreparedStatement accPs = con.prepareStatement(accountQuery))
                    {
                        accPs.setLong(1, accountNumber);
                        accPs.setString(2, email);
                        accPs.setDouble(3, 1000.00); // Default balance

                        int accountInserted = accPs.executeUpdate();
                        if (accountInserted > 0)
                        {
                            con.commit(); // Commit transaction
                            response.sendRedirect("login.jsp"); // Redirect to login
                        } else
                        {
                            con.rollback();
                            response.sendRedirect("register.jsp?error=Account creation failed");
                        }
                    }
                } else
                {
                    con.rollback();
                    response.sendRedirect("register.jsp?error=Registration failed");
                }
            }
        } catch (Exception e)
        {
            e.printStackTrace();
            response.getWriter().println("Error: " + e.getMessage());
        }
    }

    // Generate a unique 13-digit account number
    private long generateUniqueAccountNumber(Connection con) throws SQLException
    {
        long accountNumber;
        do
        {
            accountNumber = (long) (Math.random() * 9_000_000_000_000L) + 1_000_000_000_000L;
        } while (isAccountNumberExists(con, accountNumber));
        return accountNumber;
    }

    // Check if the generated account number already exists
    private boolean isAccountNumberExists(Connection con, long accountNumber) throws SQLException
    {
        String query = "SELECT 1 FROM accounts WHERE account_number = ?";
        try (PreparedStatement ps = con.prepareStatement(query))
        {
            ps.setLong(1, accountNumber);
            try (ResultSet rs = ps.executeQuery())
            {
                return rs.next();
            }
        }
```

Deposit servlet:

```java
package servlet;

import java.io.IOException;


@WebServlet("/DepositServlet")
public class DepositServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        int accountNumber = (int) session.getAttribute("account_number");  // Get account number from session
        BigDecimal amount = new BigDecimal(request.getParameter("amount"));

        try (Connection con = DatabaseConnection.getConnection()) {
            // Update balance
            String updateBalance = "UPDATE users SET balance = balance + ? WHERE account_number = ?";
            try (PreparedStatement ps = con.prepareStatement(updateBalance)) {
                ps.setBigDecimal(1, amount);
                ps.setInt(2, accountNumber);
                ps.executeUpdate();
            }

            // Insert transaction
            String insertTransaction = "INSERT INTO transactions (account_number, transaction_type, amount) VALUES (?, 'Deposit', ?)";
            try (PreparedStatement ps = con.prepareStatement(insertTransaction)) {
                ps.setInt(1, accountNumber);
                ps.setBigDecimal(2, amount);
                ps.executeUpdate();
            }

            response.sendRedirect("dashboard.jsp");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Withdraw Servlet:

```java
package servlet;
import java.io.IOException;

@WebServlet("/WithdrawServlet")
public class WithdrawServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession(false);
        // ☑ Check if user is logged in
        if (session == null || session.getAttribute("account_number") == null) {
            response.sendRedirect("login.jsp");
            return;
        }
        String accountNumber = (String) session.getAttribute("account_number"); // ☑ Using String (VARCHAR(13))
        BigDecimal amount = new BigDecimal(request.getParameter("amount").trim());
        String transactionType = "Withdrawal";
        try (Connection con = DatabaseConnection.getConnection()) {
            // ☑ Check if user has sufficient balance
            String checkBalanceQuery = "SELECT balance FROM users WHERE account_number = ?";
            try (PreparedStatement psCheck = con.prepareStatement(checkBalanceQuery)) {
                psCheck.setString(1, accountNumber);
                try (ResultSet rs = psCheck.executeQuery()) {
                    if (rs.next()) {
                        BigDecimal currentBalance = rs.getBigDecimal("balance");
                        if (currentBalance.compareTo(amount) >= 0) { // ☑ Sufficient balance
                            con.setAutoCommit(false); // ☑ Start transaction
                            // ☑ Update user's balance
                            String updateBalanceQuery = "UPDATE users SET balance = balance - ? WHERE account_number = ?";
                            try (PreparedStatement psUpdate = con.prepareStatement(updateBalanceQuery)) {
                                psUpdate.setBigDecimal(1, amount);
                                psUpdate.setString(2, accountNumber);
                                psUpdate.executeUpdate();
                            }
                            // ☑ Insert transaction record
                            String insertTransactionQuery = "INSERT INTO transactions (account_number, transaction_type, amount, transaction_date) VALUES (?, ?, ?, NOW())";
                            try (PreparedStatement ps = con.prepareStatement(insertTransactionQuery)) {
                                ps.setString(1, accountNumber);
                                ps.setString(2, transactionType);
                                ps.setBigDecimal(3, amount);
                                ps.executeUpdate();
                            }
                            con.commit(); // ☑ Commit transaction
                            session.setAttribute("message", "Withdrawal successful!");
                        } else {
                            session.setAttribute("error", "Insufficient balance.");
                        }
                    } else {
                        session.setAttribute("error", "Account not found.");
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
            session.setAttribute("error", "Transaction failed. Please try again.");
        }
        response.sendRedirect("dashboard.jsp");
    }
}
```

Transfer Servlet:

```java
package servlet;
import java.io.IOException;

@WebServlet("/TransferServlet")
public class TransferServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        int senderAccount = (int) session.getAttribute("account_number");
        int recipientAccount = Integer.parseInt(request.getParameter("recipient_account"));
        BigDecimal amount = new BigDecimal(request.getParameter("amount"));
        try (Connection con = DatabaseConnection.getConnection()) {
            // Check sender balance
            String checkBalance = "SELECT balance FROM users WHERE account_number = ?";
            PreparedStatement psCheck = con.prepareStatement(checkBalance);
            psCheck.setInt(1, senderAccount);
            java.sql.ResultSet rs = psCheck.executeQuery();
            if (rs.next()) {
                BigDecimal senderBalance = rs.getBigDecimal("balance");

                if (senderBalance.compareTo(amount) >= 0) { // Sufficient balance
                    // Deduct amount from sender
                    String deductBalance = "UPDATE users SET balance = balance - ? WHERE account_number = ?";
                    try (PreparedStatement psDeduct = con.prepareStatement(deductBalance)) {
                        psDeduct.setBigDecimal(1, amount);
                        psDeduct.setInt(2, senderAccount);
                        psDeduct.executeUpdate();
                    }

                    // Add amount to recipient
                    String addBalance = "UPDATE users SET balance = balance + ? WHERE account_number = ?";
                    try (PreparedStatement psAdd = con.prepareStatement(addBalance)) {
                        psAdd.setBigDecimal(1, amount);
                        psAdd.setInt(2, recipientAccount);
                        psAdd.executeUpdate();
                    }

                    // Insert transaction
                    String insertTransaction = "INSERT INTO transactions (account_number, transaction_type, amount) VALUES (?, 'Transfer', ?)";
                    try (PreparedStatement psSenderTransaction = con.prepareStatement(insertTransaction)) {
                        psSenderTransaction.setInt(1, senderAccount);
                        psSenderTransaction.setBigDecimal(2, amount.negate()); // Negative for sender
                        psSenderTransaction.executeUpdate();
                    }

                    try (PreparedStatement psRecipientTransaction = con.prepareStatement(insertTransaction)) {
                        psRecipientTransaction.setInt(1, recipientAccount);
                        psRecipientTransaction.setBigDecimal(2, amount); // Positive for recipient
                        psRecipientTransaction.executeUpdate();
                    }
                }
            }
            response.sendRedirect("dashboard.jsp");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Update profile servlet:

```java
package servlet;
import java.io.IOException;

@WebServlet("/UpdateProfileServlet")
public class UpdateProfileServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession(false);
        if (session == null || session.getAttribute("account_number") == null) {
            response.sendRedirect("login.jsp");
            return;
        }
        String accNumberStr = (String) session.getAttribute("account_number");
        long accNumber = Long.parseLong(accNumberStr);
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String contact = request.getParameter("contact");
        String address = request.getParameter("address");
        Connection con = null;
        PreparedStatement ps = null;
        try {
            con = dao.DatabaseConnection.getConnection();
            String updateQuery = "UPDATE users SET name=?, email=?, contact=?, address=? WHERE account_number=?";
            ps = con.prepareStatement(updateQuery);
            ps.setString(1, name);
            ps.setString(2, email);
            ps.setString(3, contact);
            ps.setString(4, address);
            ps.setLong(5, accNumber);
            int rowsUpdated = ps.executeUpdate();
            if (rowsUpdated > 0) {
                session.setAttribute("name", name); // Update session value
                response.sendRedirect("profile.jsp?success=Profile Updated");
            } else {
                response.sendRedirect("profile.jsp?error=Update Failed");
            }
        } catch (Exception e) {
            e.printStackTrace();
            response.sendRedirect("profile.jsp?error=Database Error");
        } finally {
            try { if (ps != null) ps.close(); } catch (Exception e) {}
            try { if (con != null) con.close(); } catch (Exception e) {}
        }
    }
```

Update account servlet:

```java
package servlet;

import java.io.IOException;

@WebServlet("/UpdateAccountServlet")
public class UpdateAccountServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        int userId = Integer.parseInt(request.getParameter("user_id"));
        String name = request.getParameter("name");
        String address = request.getParameter("address");
        String contact = request.getParameter("contact");

        try {
            Connection con = DatabaseConnection.getConnection();
            PreparedStatement ps = con.prepareStatement("UPDATE users SET name=?, address=?, contact=? WHERE user_id=?");
            ps.setString(1, name);
            ps.setString(2, address);
            ps.setString(3, contact);
            ps.setInt(4, userId);

            int updated = ps.executeUpdate();
            if (updated > 0) {
                response.sendRedirect("updateAccount.jsp?status=success");
            } else {
                response.sendRedirect("updateAccount.jsp?status=error");
            }
        } catch (Exception e) {
            e.printStackTrace();
            response.sendRedirect("updateAccount.jsp?status=error");
        }
    }
}
```

Index.jsp:

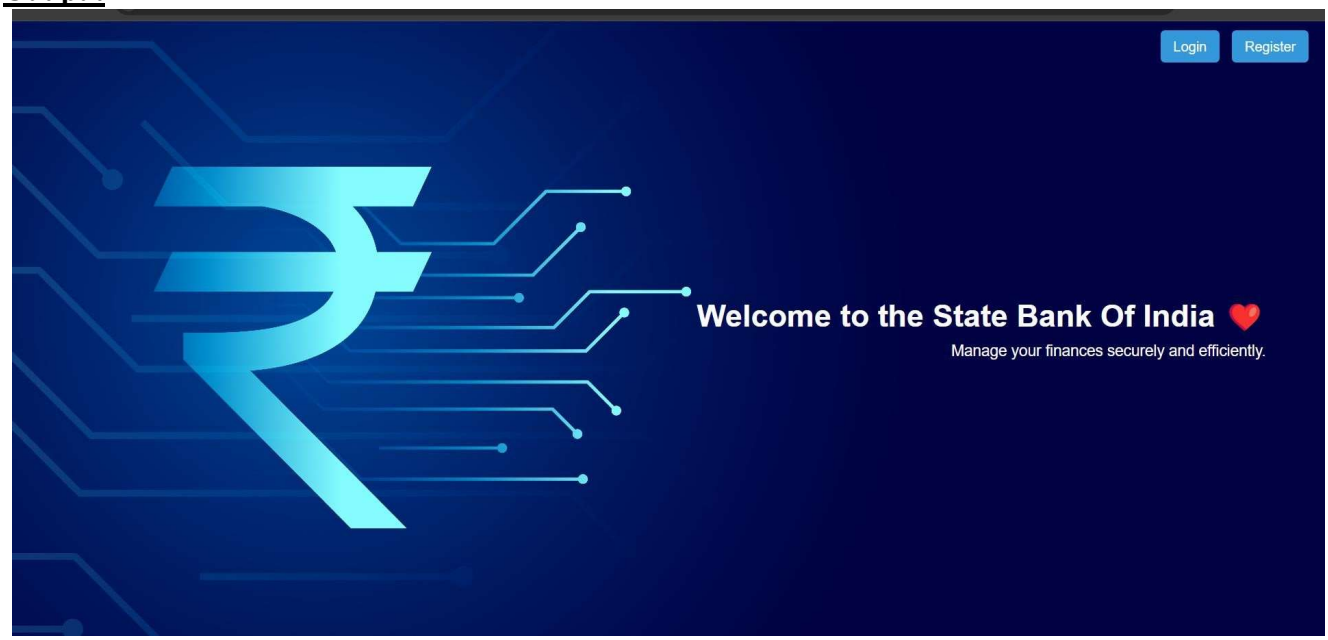
Banking System - BankingSystem/src/main/webapp/index.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

index.jsp × WithdrawSer... RegisterSer... TransferSer... UpdateAccou... UpdateProfil...

```jsp
<%@ page contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Banking System</title>
<style type="text/css">
/* styles.css */
body {
    font-family: Arial, sans-serif;
    background: url('bg-image.jpg') no-repeat center center/cover;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    text-align: center;
    margin: 0;
    padding: 0;
    color: white;
}

.container {
    padding: 20px;
    border-radius: 10px;
    width: 50%;
    text-align: right; /* Align text to the right */
    align-self: flex-end; /* Move the container to the right */
    margin-right: 50px; /* Add spacing from the right edge */
}

h1 {
    font-size: 36px;
    margin: 0;
}

p {
    font-size: 18px;
    margin-top: 10px;
}

/* Login & Register options */
.nav-links {
    position: absolute;
    top: 20px;
    right: 20px;
}

.nav-links a {
    text-decoration: none;
    color: white;
    background: #3498db;
    padding: 10px 15px;
    border-radius: 5px;
    margin-left: 10px;
    transition: background 0.3s ease-in-out;
}

.nav-links a:hover {
    background: #2980b9;
}
</style>
</head>
<body>
    <!-- Navigation Links -->
    <div class="nav-links">
        <a href="login.jsp">Login</a> <a href="register.jsp">Register</a>
    </div>

    <!-- Welcome Message Container -->
    <div class="container">
        <h1>Welcome to the State Bank Of India ♥</h1>
        <p>Manage your finances securely and efficiently.</p>
    </div>
</body>
</html>
```

Out put



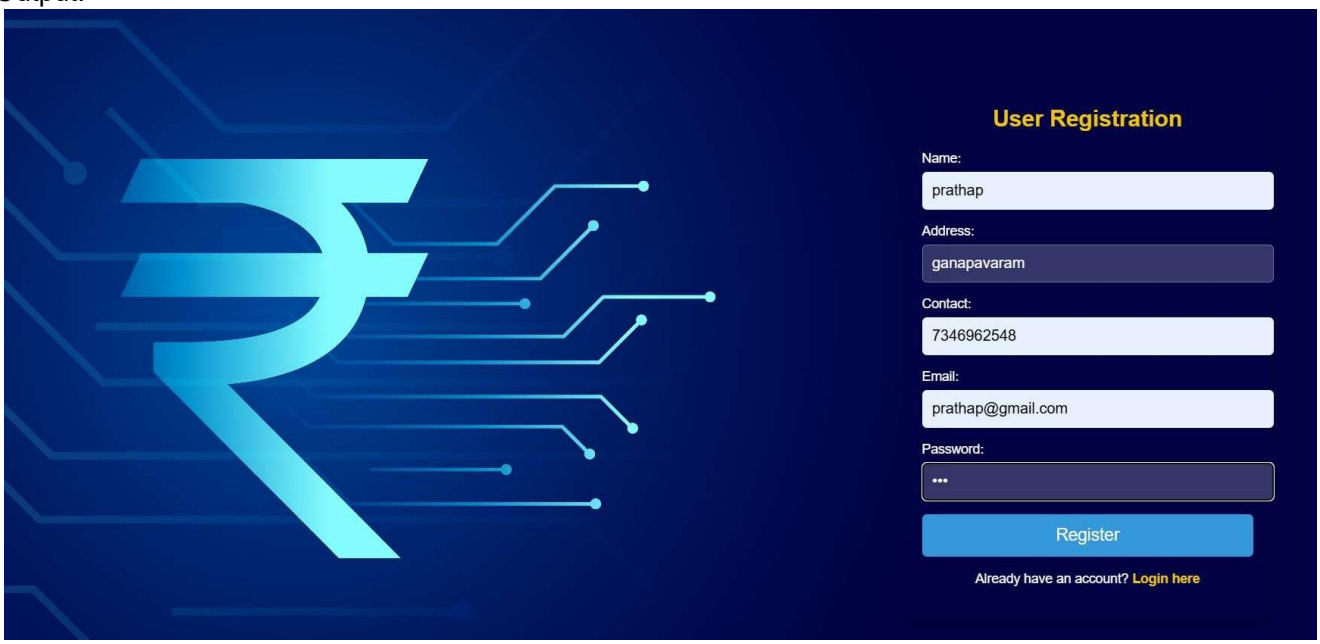
Registration.jsp:

```
<%@ page contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>User Registration</title>
<style>
/* Global Styles */
body {
    font-family: Arial, sans-serif;
    background: url('bg-image.jpg') no-repeat center center/cover;
    height: 100vh;
    display: flex;
    justify-content: flex-end; /* Moves content to the right */
    align-items: center;
    margin: 0;
    padding-right: 10%;
    color: white;
}

/* Form Container - No Background */
.container {
    width: 350px;
    text-align: center;
    padding: 20px;
    border-radius: 15px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); /* Soft shadow */
    backdrop-filter: blur(10px); /* Glass effect */
    -webkit-backdrop-filter: blur(10px); /* Safari Support */
    border: none; /* Removed border */
}

h2 {
    font-size: 24px;
    color: #f1c40f;
}

/* Input Fields */
.input-group {
    margin-bottom: 15px;
    text-align: left;
}

.input-group label {
    display: block;
    font-size: 14px;
    margin-bottom: 5px;
}

.input-group input {
    width: 100%;
    padding: 10px;
    border: 1px solid rgba(255, 255, 255, 0.3);
    border-radius: 5px;
    font-size: 16px;
    background: rgba(255, 255, 255, 0.2); /* Light transparent white */
```

```
    font-size: 16px;
    background: rgba(255, 255, 255, 0.2); /* Light transparent white */
    color: white;
}

.input-group input::placeholder {
    color: rgba(255, 255, 255, 0.7);
}

/* Submit Button */
.btn {
    background: #3498db;
    color: white;
    padding: 12px;
    width: 100%;
    border: none;
    border-radius: 5px;
    font-size: 18px;
    cursor: pointer;
    transition: background 0.3s;
}

.btn:hover {
    background: #2980b9;
}

/* Login Link */
.login-link {
    margin-top: 15px;
    font-size: 14px;
}

.login-link a {
    color: #f1c40f;
    text-decoration: none;
    font-weight: bold;
}

.login-link a:hover {
    text-decoration: underline;
}
</style>
</head>
<body>

    <div class="container">
        <h2>User Registration</h2>
        <form action="RegisterServlet" method="post">

            <div class="input-group">
                <label for="name">Name:</label> <input type="text" name="name"
                    id="name" required placeholder="Enter your name">
            </div>

            <div class="input-group">
                <label for="address">Address:</label> <input type="text"
                    name="address" id="address" placeholder="Enter your address">
```

```
            <div class="input-group">
                <label for="address">Address:</label> <input type="text"
                    name="address" id="address" placeholder="Enter your address">
            </div>

            <div class="input-group">
                <label for="contact">Contact:</label> <input type="text"
                    name="contact" id="contact" required
                    placeholder="Enter your contact number">
            </div>

            <div class="input-group">
                <label for="email">Email:</label> <input type="email" name="email"
                    id="email" required placeholder="Enter your email">
            </div>

            <div class="input-group">
                <label for="password">Password:</label> <input type="password"
                    name="password" id="password" required
                    placeholder="Enter your password">
            </div>


            <input type="submit" class="btn" value="Register">
        </form>

        <!-- Login Link -->
        <p class="login-link">
            Already have an account? <a href="login.jsp">Login here</a>
        </p>
    </div>

</body>
</html>
```

Output:


User Registration
Name:
prathap
Address:
ganapavaram
Contact:
7346962548
Email:
prathap@gmail.com
Password:
•••
Register
Already have an account? Login here

index.jsp | RegisterSer... | TransferSer... | UpdateAccou... | UpdateProfil... | login.jsp

```jsp
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Login</title>
    <style>
        /* Global Styles */
        body {
            font-family: Arial, sans-serif;
            background: url('bg-image.jpg') no-repeat center center/cover;
            height: 100vh;
            display: flex;
            justify-content: flex-end; /* Moves content to the right */
            align-items: center;
            margin: 0;
            padding-right: 10%;
            color: white;
        }

        /* Form Container - Glassmorphism Effect */
        .container {
            width: 350px;
            text-align: center;
            padding: 20px;
            border-radius: 15px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); /* Soft shadow */
            backdrop-filter: blur(10px); /* Glass effect */
            -webkit-backdrop-filter: blur(10px); /* Safari Support */
            border: none; /* No border */
        }

        h2 {
            font-size: 24px;
            color: #f1c40f;
        }

        /* Input Fields */
        .input-group {
            margin-bottom: 15px;
            text-align: left;
        }

        .input-group label {
            display: block;
            font-size: 14px;
            margin-bottom: 5px;
        }

        .input-group input {
            width: 100%;
            padding: 10px;
            border: 1px solid rgba(255, 255, 255, 0.3);
            border-radius: 5px;
            font-size: 16px;
            background: rgba(255, 255, 255, 0.2); /* Light transparent white */
            color: white;
        }

        .input-group input::placeholder {
            color: rgba(255, 255, 255, 0.7);
        }

        /* Submit Button */
        .btn {
            background: #3498db;
            color: white;
            padding: 12px;
            width: 100%;
            border: none;
            border-radius: 5px;
            font-size: 18px;
            cursor: pointer;
            transition: background 0.3s;
        }

        .btn:hover {
            background: #2980b9;
        }

        /* Register Link */
        .register-link {
            margin-top: 15px;
            font-size: 14px;
```

```html
            width: 100%;
            border: none;
            border-radius: 5px;
            font-size: 18px;
            cursor: pointer;
            transition: background 0.3s;
        }

        .btn:hover {
            background: #2980b9;
        }

        /* Register Link */
        .register-link {
            margin-top: 15px;
            font-size: 14px;
        }

        .register-link a {
            color: #f1c40f;
            text-decoration: none;
            font-weight: bold;
        }

        .register-link a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>

    <div class="container">
        <h2>User Login</h2>
        <form action="loginServlet" method="post">

            <div class="input-group">
                <label for="emailOrContact">Email or Contact:</label>
                <input type="text" name="emailOrContact" id="emailOrContact" required placeholder="Enter your email or contact">
            </div>

            <div class="input-group">
                <label for="password">Password:</label>
                <input type="password" name="password" id="password" required placeholder="Enter your password">
            </div>

            <input type="submit" class="btn" value="Login">
        </form>

        <!-- Register Link -->
        <p class="register-link">
            Don't have an account? <a href="register.jsp">Register here</a>
        </p>
    </div>

</body>
</html>
```

Output:

Dashboard.jsp:

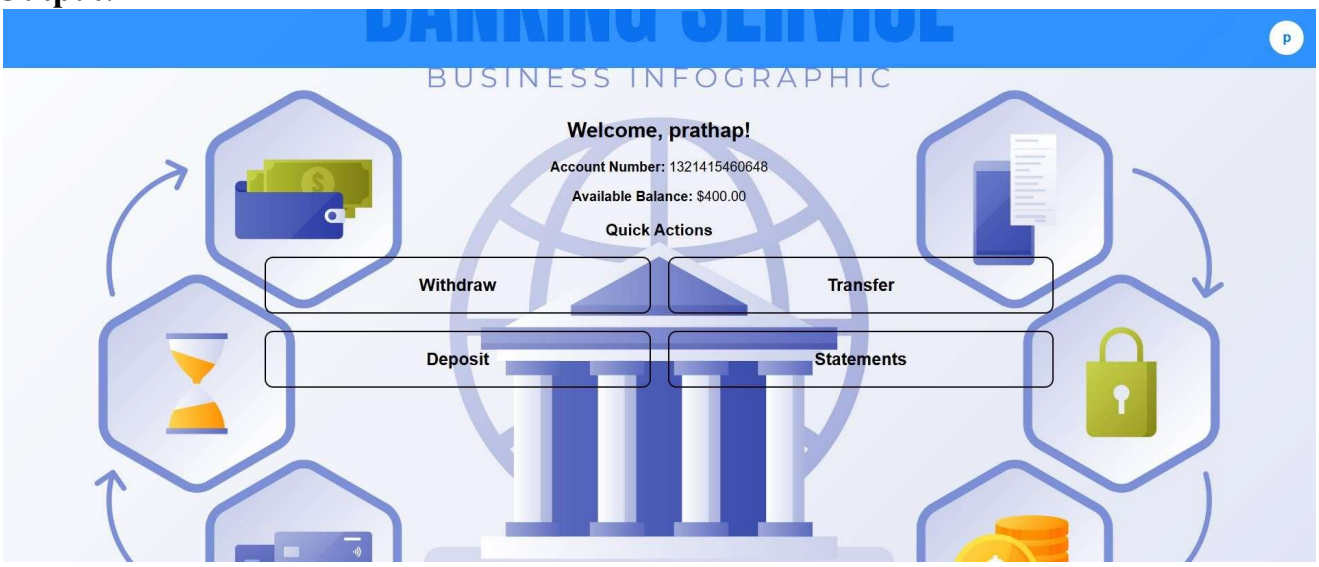
```jsp
<%@ page import="java.sql.*"%>
<%@ page import="java.text.DecimalFormat"%>
<%@ page import="java.util.*"%>

<%
HttpSession userSession = request.getSession(false);
if (userSession == null || userSession.getAttribute("account_number") == null)
{
    response.sendRedirect("login.jsp"); // Redirect to login if session is invalid
    return;
}

// Database connection
Connection con = null;
PreparedStatement ps = null;
ResultSet rs = null;

try
{
    con = dao.DatabaseConnection.getConnection();

    // Retrieve user details from session
    String name = (String) userSession.getAttribute("name");
    String accNumberStr = (String) userSession.getAttribute("account_number");
    long accNumber = Long.parseLong(accNumberStr);

    // Fetch available balance
    String balanceQuery = "SELECT balance FROM accounts WHERE account_number=?";
    ps = con.prepareStatement(balanceQuery);
    ps.setLong(1, accNumber);
    rs = ps.executeQuery();

    double balance = 0;
    if (rs.next())
    {
        balance = rs.getDouble("balance");
    }
    rs.close();
    ps.close();
%>

<!DOCTYPE html>
<html>
<head>
<title>Dashboard</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('dashboard.jpg') no-repeat center center fixed;
    background-size: cover;
    text-align: center;
    margin: 0;
    padding: 0;
    color: black; /* Ensure text remains black */
}

.navbar {
```

```
navbar {
    background: rgba(0, 123, 255, 0.8);
    padding: 15px;
    display: flex;
    justify-content: flex-end;
    align-items: center;
    position: relative;

profile-container {
    position: relative;
    display: inline-block;
    cursor: pointer;

profile-icon {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    background: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: bold;
    color: #007bff;

dropdown-content {
    display: none;
    position: absolute;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
    border-radius: 5px;
    min-width: 120px;
    text-align: left;

dropdown-content a {
    color: white;
    padding: 10px;
    display: block;
    text-decoration: none;

dropdown-content a:hover {
    background: rgba(255, 255, 255, 0.2);

profile-container:hover .dropdown-content {
    display: block;

container {
    width: 60%;
    margin: 20px auto;
    padding: 20px;
```

```
    margin: 20px auto;
    padding: 20px;
    border-radius: 10px;
    color: black; /* Ensure all text is black */
}

.grid-container {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 20px;
    margin-top: 20px;
    color: black;
}

.grid-item {
    background: transparent;
    color: black;
    padding: 20px;
    border-radius: 10px;
    font-size: 20px;
    font-weight: bold;
    text-decoration: none;
    display: flex;
    align-items: center;
    justify-content: center;
    transition: transform 0.2s, background 0.3s;
    border: 2px solid black; /* Updated to black */
}

.grid-item:hover {
    background: rgba(0, 0, 0, 0.1);
    transform: scale(1.05);
}
</style>
</head>
<body>

    <!-- Navbar with Profile Dropdown -->
    <div class="navbar">
        <div class="profile-container">
            <div class="profile-icon">&lt;%=name.charAt(0)%&gt;</div>
            <div class="dropdown-content">
                <a href="profile.jsp">Profile</a> <a href="index.jsp">Logout</a>
            </div>
        </div>
    </div>

    <div class="container">
        <h2>
            Welcome,
            <%=name%>!
        </h2>
        <p>
            <strong>Account Number:</strong>
            <%=accNumber%></p>
        <p>
            <strong>Available Balance:</strong> $<%=new DecimalFormat("#,##0.00").format(balance)%></p>
```

```
            <%=accNumber%></p>
        <p>
            <strong>Available Balance:</strong> $<%=new DecimalFormat("#,##0.00").format(balance)%></p>

        <h3>Quick Actions</h3>
        <div class="grid-container">
            <a href="withdraw.jsp" class="grid-item">Withdraw</a> <a
                href="transfer.jsp" class="grid-item">Transfer</a> <a
                href="deposit.jsp" class="grid-item">Deposit</a> <a
                href="statements.jsp" class="grid-item">Statements</a>
        </div>
    </div>

</body>
</html>

<%
} catch (Exception e)
{
e.printStackTrace();
out.println("<p style='color:red;'>Error fetching account details.</p>");
} finally
{
if (rs != null)
    try
    {
        rs.close();
    } catch (SQLException e)
    {
    }
if (ps != null)
    try
    {
        ps.close();
    } catch (SQLException e)
    {
    }
if (con != null)
    try
    {
        con.close();
    } catch (SQLException e)
    {
    }
}
%>
```

Output:



Deposit.jsp:

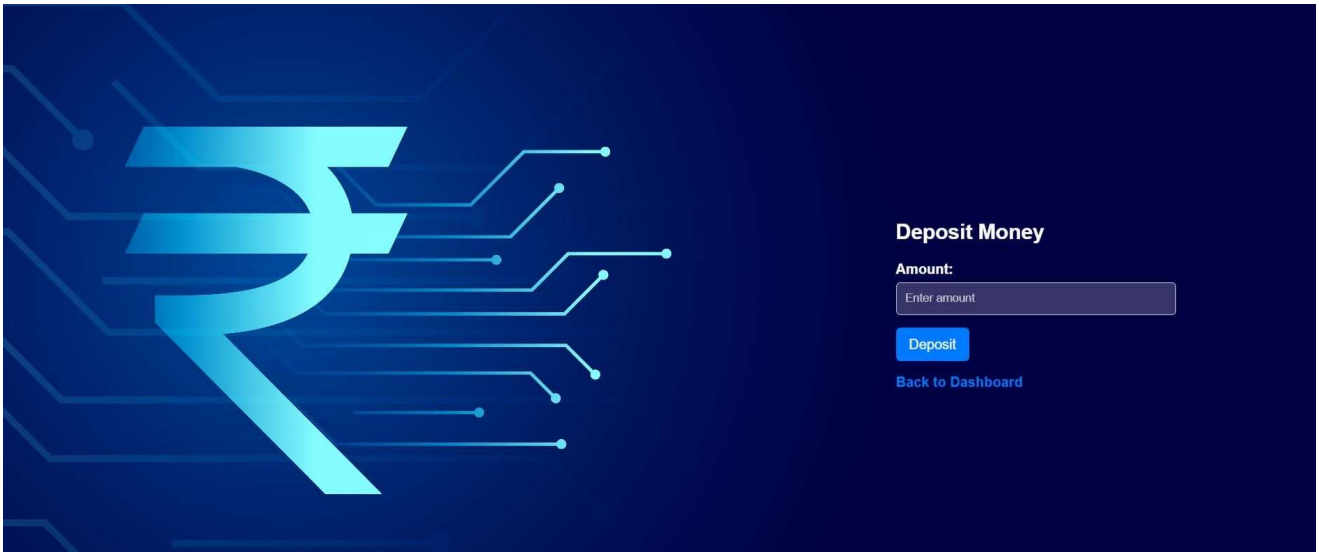
```
<%@ page import="java.sql.*"%>
<%
HttpSession sessionObj = request.getSession(false);
if (sessionObj == null || sessionObj.getAttribute("account_number") == null)
{
    response.sendRedirect("login.jsp");
    return;
}

String message = "";
if (request.getMethod().equalsIgnoreCase("POST"))
{
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try
    {
        con = dao.DatabaseConnection.getConnection();
        String accNumber = (String) sessionObj.getAttribute("account_number");
        double amount = Double.parseDouble(request.getParameter("amount"));

        // Check if account exists
        PreparedStatement checkAccount = con.prepareStatement("SELECT balance FROM accounts WHERE account_number = ?");
        checkAccount.setString(1, accNumber);
        rs = checkAccount.executeQuery();

        if (!rs.next())
        {
    message = "<p class='error'>Error: Account does not exist.</p>";
        } else
        {
    con.setAutoCommit(false); // Start transaction

    // Update balance
    ps = con.prepareStatement("UPDATE accounts SET balance = balance + ? WHERE account_number = ?");
    ps.setDouble(1, amount);
    ps.setString(2, accNumber);
    int updatedRows = ps.executeUpdate();

    if (updatedRows > 0)
    {
        // Insert transaction record
        PreparedStatement transaction = con.prepareStatement(
                "INSERT INTO transactions(account_number, transaction_type, amount) VALUES(?, 'Deposit', ?)");
        transaction.setString(1, accNumber);
        transaction.setDouble(2, amount);
        transaction.executeUpdate();

        con.commit(); // Commit transaction
        message = "<p class='success'>Deposit successful!</p>";
```

```
        con.commit(); // Commit transaction
        message = "<p class='success'>Deposit successful!</p>";
    } else
    {
        message = "<p class='error'>Error updating balance.</p>";
    }
    con.setAutoCommit(true);
        }
    } catch (Exception e)
    {
        if (con != null)
    con.rollback(); // Rollback on failure
        e.printStackTrace();
        message = "<p class='error'>Transaction failed.</p>";
    } finally
    {
        if (rs != null)
    rs.close();
        if (ps != null)
    ps.close();
        if (con != null)
    con.close();
    }
}
%>

<!DOCTYPE html>
<html>
<head>
<title>Deposit Money</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('bg-image.jpg') no-repeat center center fixed;
    background-size: cover;
    margin: 0;
    padding: 0;
    color: white;
}

.navbar {
    background: rgba(0, 123, 255, 0.8);
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.profile-container {
    margin-right: 20px;
}
```

```
.profile-container {
    margin-right: 20px;
}

.profile-icon {
    color: white;
    font-weight: bold;
    cursor: pointer;
}

.container {
    position: absolute;
    top: 50%;
    right: 10%;
    transform: translateY(-50%);
    padding: 30px;
    text-align: left;
    border-radius: 10px;
    width: 300px;
    backdrop-filter: blur(10px); /* Adds a smooth blur effect */
    -webkit-backdrop-filter: blur(10px);
}

h2 {
    margin-bottom: 20px;
    color: white;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background: rgba(255, 255, 255, 0.2);
    color: white;
}

input::placeholder {
    color: rgba(255, 255, 255, 0.7);
}

button {
    background: #007bff;
```

```
    background: #007bff;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background: #0056b3;
}

.back-button {
    display: block;
    margin-top: 15px;
    color: #007bff;
    text-decoration: none;
    font-weight: bold;
}

.back-button:hover {
    text-decoration: underline;
}

.success {
    color: lightgreen;
    font-weight: bold;
}

.error {
    color: red;
    font-weight: bold;
}
</style>
</head>
<body>


    <div class="container">
        <h2>Deposit Money</h2>
        <form method="post">
            <label>Amount:</label> <input type="number" name="amount" step="0.01"
                placeholder="Enter amount" required>
            <button type="submit">Deposit</button>
        </form>
        <%=message%>
        <a href="dashboard.jsp" class="back-button">Back to Dashboard</a>
    </div>

</body>
```

Output:



Withdraw:

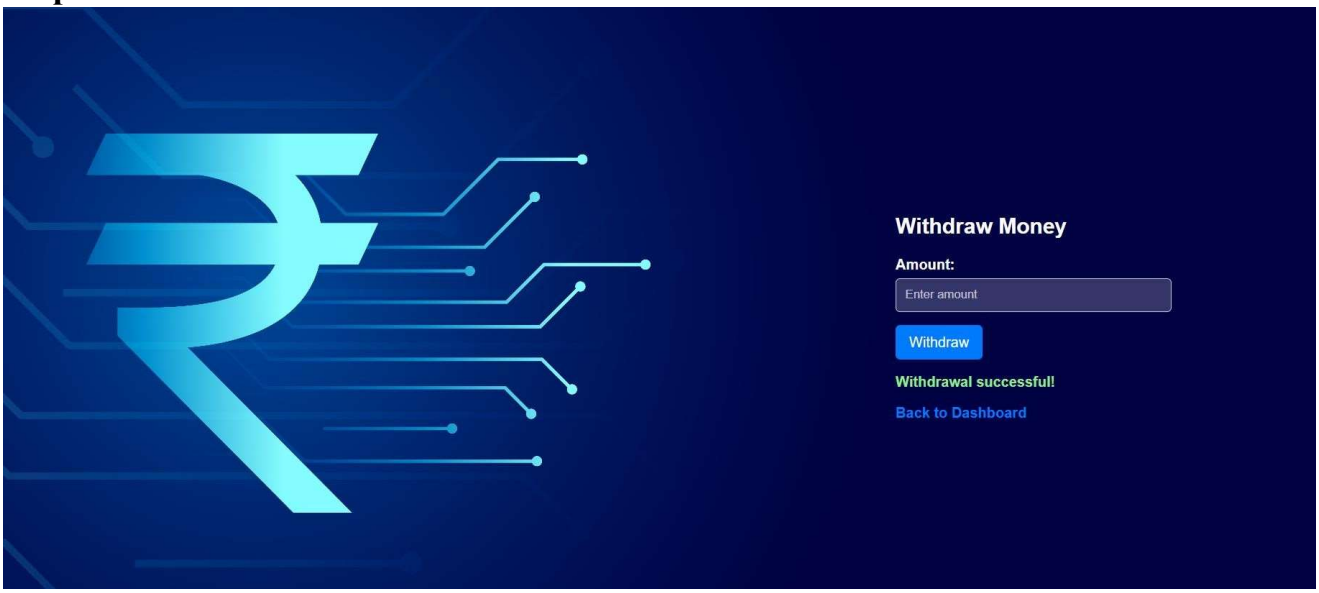
```
<%@ page import="java.sql.*"%>
<%
HttpSession sessionObj = request.getSession(false);
if (sessionObj == null || sessionObj.getAttribute("account_number") == null)
{
    response.sendRedirect("login.jsp");
    return;
}

String message = "";
if (request.getMethod().equalsIgnoreCase("POST"))
{
    Connection con = null;
    PreparedStatement ps = null;

    try
    {
        con = dao.DatabaseConnection.getConnection();
        String accNumber = (String) sessionObj.getAttribute("account_number");
        double amount = Double.parseDouble(request.getParameter("amount"));

        // Check account balance
        PreparedStatement checkBalance = con.prepareStatement("SELECT balance FROM accounts WHERE account_number=?");
        checkBalance.setString(1, accNumber);
        ResultSet rs = checkBalance.executeQuery();

        if (rs.next() && rs.getDouble("balance") >= amount)
        {
    // Update balance
    ps = con.prepareStatement("UPDATE accounts SET balance = balance - ? WHERE account_number=?");
    ps.setDouble(1, amount);
    ps.setString(2, accNumber);
    ps.executeUpdate();

    // Record transaction
    PreparedStatement transaction = con.prepareStatement(
            "INSERT INTO transactions(account_number, transaction_type, amount) VALUES(?, 'Withdraw', ?)");
    transaction.setString(1, accNumber);
    transaction.setDouble(2, amount);
    transaction.executeUpdate();

    message = "<p class='success'>Withdrawal successful!</p>";
        } else
        {
    message = "<p class='error'>Insufficient balance!</p>";
        }
    } catch (Exception e)
    {
        e.printStackTrace();
        message = "<p class='error'>Transaction failed.</p>";
    }
```

```
    }
}
%>

<!DOCTYPE html>
<html>
<head>
<title>Withdraw Money</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('bg-image.jpg') no-repeat center center fixed;
    background-size: cover;
    margin: 0;
    padding: 0;
    color: white;
}

.navbar {
    background: rgba(0, 123, 255, 0.8);
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.profile-container {
    margin-right: 20px;
}

.profile-icon {
    color: white;
    font-weight: bold;
    cursor: pointer;
}

.container {
    position: absolute;
    top: 50%;
    right: 10%;
    transform: translateY(-50%);
    padding: 30px;
    text-align: left;
    background: transparent; /* Fully transparent */
    border-radius: 10px;
    box-shadow: none; /* Removes any shadow */
}


h2 {
    margin-bottom: 20px;
```

```
h2 {
    margin-bottom: 20px;
    color: white;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background: rgba(255, 255, 255, 0.2);
    color: white;
}

input::placeholder {
    color: rgba(255, 255, 255, 0.7);
}

button {
    background: #007bff;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background: #0056b3;
}

.back-button {
    display: block;
    margin-top: 15px;
    color: #007bff;
    text-decoration: none;
    font-weight: bold;
}

.back-button:hover {
    text-decoration: underline;
}
```

```
}

.back-button:hover {
    text-decoration: underline;
}

.success {
    color: lightgreen;
    font-weight: bold;
}

.error {
    color: red;
    font-weight: bold;
}
</style>
</head>
<body>

    <div class="container">
        <h2>Withdraw Money</h2>
        <form method="post">
            <label>Amount:</label> <input type="number" name="amount" required
                placeholder="Enter amount">
            <button type="submit">Withdraw</button>
        </form>
        <%=message%>
        <a href="dashboard.jsp" class="back-button">Back to Dashboard</a>
    </div>

</body>
</html>
```

Output:



Transfer.jsp:

```
<%@ page import="java.sql.*"%>
<%
HttpSession sessionObj = request.getSession(false);
if (sessionObj == null || sessionObj.getAttribute("account_number") == null)
{
    response.sendRedirect("login.jsp");
    return;
}

String message = "";
if (request.getMethod().equalsIgnoreCase("POST"))
{
    Connection con = null;
    PreparedStatement ps = null;

    try
    {
        con = dao.DatabaseConnection.getConnection();
        String senderAcc = (String) sessionObj.getAttribute("account_number");
        String receiverAcc = request.getParameter("receiver");
        double amount = Double.parseDouble(request.getParameter("amount"));

        // Check sender's balance
        PreparedStatement checkBalance = con.prepareStatement("SELECT balance FROM accounts WHERE account_number=?");
        checkBalance.setString(1, senderAcc);
        ResultSet rs = checkBalance.executeQuery();

        if (rs.next() && rs.getDouble("balance") >= amount)
        {
    // Deduct from sender
    ps = con.prepareStatement("UPDATE accounts SET balance = balance - ? WHERE account_number=?");
    ps.setDouble(1, amount);
    ps.setString(2, senderAcc);
    ps.executeUpdate();

    // Add to receiver
    ps = con.prepareStatement("UPDATE accounts SET balance = balance + ? WHERE account_number=?");
    ps.setDouble(1, amount);
    ps.setString(2, receiverAcc);
    ps.executeUpdate();

    // Record sender transaction
    PreparedStatement transaction1 = con.prepareStatement(
            "INSERT INTO transactions(account_number, transaction_type, amount) VALUES(?, 'Debit', ?)");
    transaction1.setString(1, senderAcc);
    transaction1.setDouble(2, amount);
    transaction1.executeUpdate();

    // Record receiver transaction
    PreparedStatement transaction2 = con.prepareStatement(
            "INSERT INTO transactions(account_number, transaction_type, amount) VALUES(?, 'Credit', ?)");
```

```
    PreparedStatement transaction2 = con.prepareStatement(
            "INSERT INTO transactions(account_number, transaction_type, amount) VALUES(?, 'Credit', ?)");
    transaction2.setString(1, receiverAcc);
    transaction2.setDouble(2, amount);
    transaction2.executeUpdate();

    message = "<p class='success'>Transfer successful!</p>";
        } else
        {
    message = "<p class='error'>Insufficient balance or invalid account!</p>";
        }
    } catch (Exception e)
    {
        e.printStackTrace();
        message = "<p class='error'>Transaction failed.</p>";
    }
}
%>

<!DOCTYPE html>
<html>
<head>
<title>Transfer Money</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('bg-image.jpg') no-repeat center center fixed;
    background-size: cover;
    margin: 0;
    padding: 0;
    color: white;
}

.navbar {
    background: rgba(0, 123, 255, 0.8);
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.profile-container {
    margin-right: 20px;
}

.profile-icon {
    color: white;
    font-weight: bold;
    cursor: pointer;
}
```

```
container {
    position: absolute;
    top: 50%;
    right: 10%;
    transform: translateY(-50%);
    padding: 30px;
    text-align: left;
    border-radius: 10px;
    width: 300px;
    backdrop-filter: blur(10px); /* Adds a smooth blur effect */
    -webkit-backdrop-filter: blur(10px);
}

h2 {
    margin-bottom: 20px;
    color: white;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background: rgba(255, 255, 255, 0.2);
    color: white;
}

input::placeholder {
    color: rgba(255, 255, 255, 0.7);
}

button {
    background: #007bff;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background: #0056b3;
}
```


```
container {
    position: absolute;
    top: 50%;
    right: 10%;
    transform: translateY(-50%);
    padding: 30px;
    text-align: left;
    border-radius: 10px;
    width: 300px;
    backdrop-filter: blur(10px); /* Adds a smooth blur effect */
    -webkit-backdrop-filter: blur(10px);
}

h2 {
    margin-bottom: 20px;
    color: white;
}

label {
    display: block;
    font-weight: bold;
    margin-bottom: 5px;
}

input {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    background: rgba(255, 255, 255, 0.2);
    color: white;
}

input::placeholder {
    color: rgba(255, 255, 255, 0.7);
}

button {
    background: #007bff;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background: #0056b3;
```

```
utton:hover {
    background: #0056b3;
}

back-button {
    display: block;
    margin-top: 15px;
    color: #007bff;
    text-decoration: none;
    font-weight: bold;
}

back-button:hover {
    text-decoration: underline;
}

success {
    color: lightgreen;
    font-weight: bold;
}

error {
    color: red;
    font-weight: bold;
}
/style>
/head>
body>

    <div class="container">
        <h2>Transfer Money</h2>
        <form method="post">
            <label>Receiver Account:</label> <input type="text" name="receiver"
                placeholder="Enter account number" required> <label>Amount:</label>
            <input type="number" name="amount" placeholder="Enter amount"
                required>
            <button type="submit">Transfer</button>
        </form>
        <%=message%>
        <a href="dashboard.jsp" class="back-button">Back to Dashboard</a>
    </div>

/body>
/html>
```

Output:

Statement.jsp:

```jsp
<%@ page import="java.sql.*"%>
<%@ page import="java.text.SimpleDateFormat"%>

<%
HttpSession userSession = request.getSession(false);
if (userSession == null || userSession.getAttribute("account_number") == null)
{
    response.sendRedirect("login.jsp");
    return;
}

// Database connection
Connection con = null;
PreparedStatement ps = null;
ResultSet rs = null;

try
{
    con = dao.DatabaseConnection.getConnection();

    // Retrieve user details from session
    String name = (String) userSession.getAttribute("name");
    String accNumberStr = (String) userSession.getAttribute("account_number");
    long accNumber = Long.parseLong(accNumberStr);

    // Fetch transaction history
    String query = "SELECT transaction_type, amount, transaction_date FROM transactions WHERE account_number=? ORDER BY transaction_date DESC";
    ps = con.prepareStatement(query);
    ps.setLong(1, accNumber);
    rs = ps.executeQuery();
%>

<!DOCTYPE html>
<html>
<head>
<title>Account Statements</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('background.jpg') no-repeat center center fixed;
    background-size: cover;
    text-align: center;
    margin: 0;
    padding: 0;
    color: white;
}

.navbar {
    background: rgba(0, 123, 255, 0.9);
    padding: 15px;
    display: flex;
```

```jsp
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    position: relative;
}

.profile-container {
    position: relative;
    cursor: pointer;
}

.profile-icon {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    background: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-weight: bold;
    color: #007bff;
}

.dropdown {
    display: none;
    position: absolute;
    right: 0;
    top: 50px;
    background: white;
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
    border-radius: 5px;
    overflow: hidden;
    text-align: left;
}

.dropdown a {
    display: block;
    padding: 10px;
    color: black;
    text-decoration: none;
}

.dropdown a:hover {
    background: #007bff;
    color: white;
}

.profile-container:hover .dropdown {
    display: block;
}
```

```
.container {
    width: 80%;
    margin: 20px auto;
}

h2 {
    color: white;
}

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
    color: white;
}

th, td {
    padding: 10px;
    text-align: center;
}

th {
    background: rgba(0, 123, 255, 0.8);
    color: white;
    border-radius: 5px;
}

tr:nth-child(even) {
    background: rgba(255, 255, 255, 0.2);
}

tr:hover {
    background: rgba(255, 255, 255, 0.3);
}

.back-btn {
    padding: 10px 20px;
    background: #28a745;
    color: white;
    text-decoration: none;
    border-radius: 5px;
    display: inline-block;
    margin-top: 20px;
}

.back-btn:hover {
    background: #218838;
}
</style>
</head>
```

```
            </tr>
            <%
            }
            if (!hasTransactions)
            {
            %>
            <tr>
                <td colspan="3">No transactions found</td>
            </tr>
            <%
            }
            %>
        </table>

        <a href="dashboard.jsp" class="back-btn">Back to Dashboard</a>
    </div>

</body>
</html>

<%
} catch (Exception e)
{
e.printStackTrace();
out.println("<p style='color:red;'>Error fetching transaction history.</p>");
} finally
{
if (rs != null)
    try
    {
        rs.close();
    } catch (SQLException e)
    {
    }
if (ps != null)
    try
    {
        ps.close();
    } catch (SQLException e)
    {
    }
if (con != null)
    try
    {
        con.close();
    } catch (SQLException e)
    {
    }
}
```

Output:

Type	Amount	Date
Withdraw	$100.0	2025-04-05 23:21:11
Debit	$100.0	2025-04-05 19:06:27
Deposit	$500.0	2025-04-05 19:05:04
Withdraw	$1000.0	2025-04-05 19:04:46

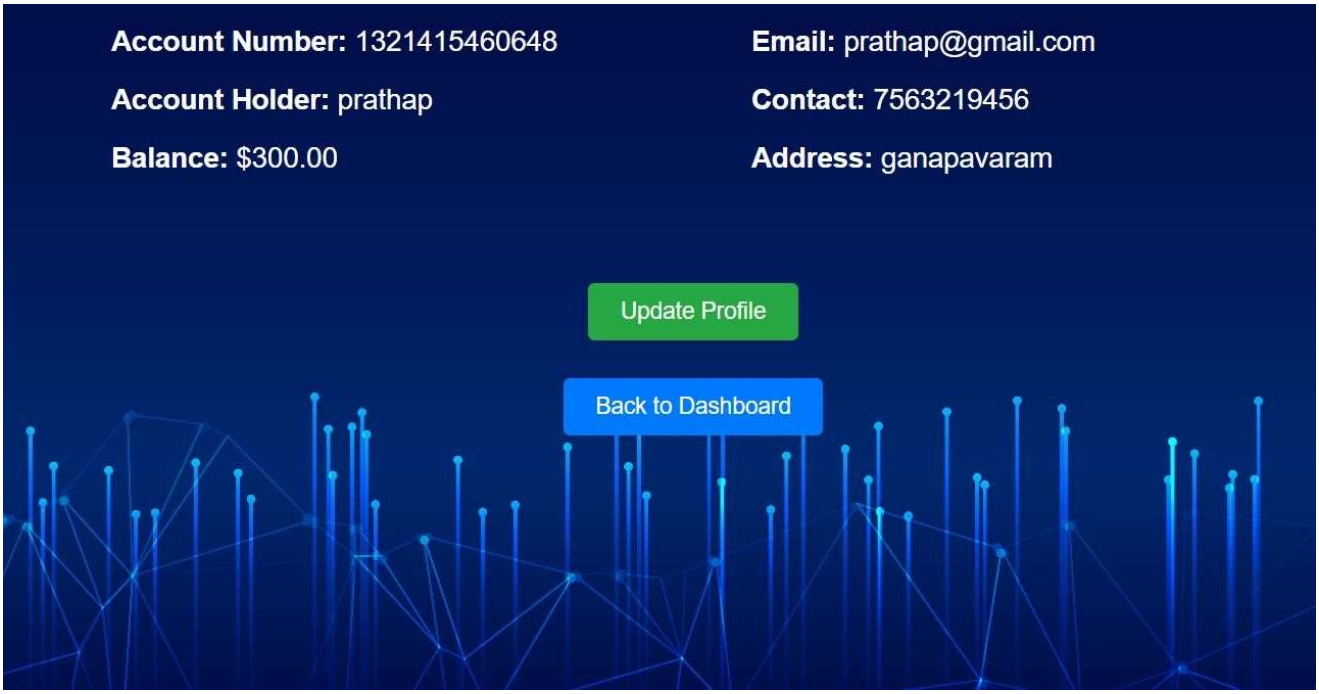
Back to Dashboard

Profile.jsp:

```
<%@ page import="java.sql.*"%>
<%
HttpSession userSession = request.getSession(false);
if (userSession == null || userSession.getAttribute("account_number") == null) {
    response.sendRedirect("login.jsp");
    return;
}

Connection con = null;
PreparedStatement ps = null;
ResultSet rs = null;

try {
    con = dao.DatabaseConnection.getConnection();
    String accNumberStr = (String) userSession.getAttribute("account_number");
    long accNumber = Long.parseLong(accNumberStr);

    // Fetch user details along with account balance
    String query = "SELECT u.name, u.email, u.contact, u.address, a.account_number, a.balance " +
                   "FROM users u JOIN accounts a ON u.account_number = a.account_number " +
                   "WHERE u.account_number=?";

    ps = con.prepareStatement(query);
    ps.setLong(1, accNumber);
    rs = ps.executeQuery();

    String name = "", email = "", contact = "", address = "";
    long accountNumber = 0;
    double balance = 0.0;

    if (rs.next()) {
        name = rs.getString("name");
        email = rs.getString("email");
        contact = rs.getString("contact");
        address = rs.getString("address");
        accountNumber = rs.getLong("account_number");
        balance = rs.getDouble("balance");
    }
%>

<!DOCTYPE html>
<html>
<head>
<title>Profile</title>
<style>
body {
    font-family: Arial, sans-serif;
    background: url('background.jpg') no-repeat center center fixed;
    background-size: cover;
    color: white;
    text-align: center;
```

```
    background: #007bff;
    color: white;
}

.dashboard-btn:hover {
    background: #0056b3;
}
</style>
</head>
<body>

    <div class="container">
        <!-- Left Section: Account Details -->
        <div class="left-section">
            <p><strong>Account Number:</strong> <%=accountNumber%></p>
            <p><strong>Account Holder:</strong> <%=name%></p>
            <p><strong>Balance:</strong> $<%=String.format("%.2f", balance)%></p>
        </div>

        <!-- Right Section: Contact Details -->
        <div class="right-section">
            <p><strong>Email:</strong> <%=email%></p>
            <p><strong>Contact:</strong> <%=contact%></p>
            <p><strong>Address:</strong> <%=address%></p>
        </div>
    </div>

    <!-- Buttons for navigation -->
    <div class="btn-container">
        <form action="updateProfile.jsp" method="get">
            <button type="submit" class="btn update-btn">Update Profile</button>
        </form>
        <form action="dashboard.jsp" method="get">
            <button type="submit" class="btn dashboard-btn">Back to Dashboard</button>
        </form>
    </div>

</body>
</html>

<%
} catch (Exception e) {
    e.printStackTrace();
    out.println("<p style='color:red;'>Error fetching profile details.</p>");
} finally {
    if (rs != null) try { rs.close(); } catch (SQLException e) {}
    if (ps != null) try { ps.close(); } catch (SQLException e) {}
    if (con != null) try { con.close(); } catch (SQLException e) {}
}
%>
```

Output:



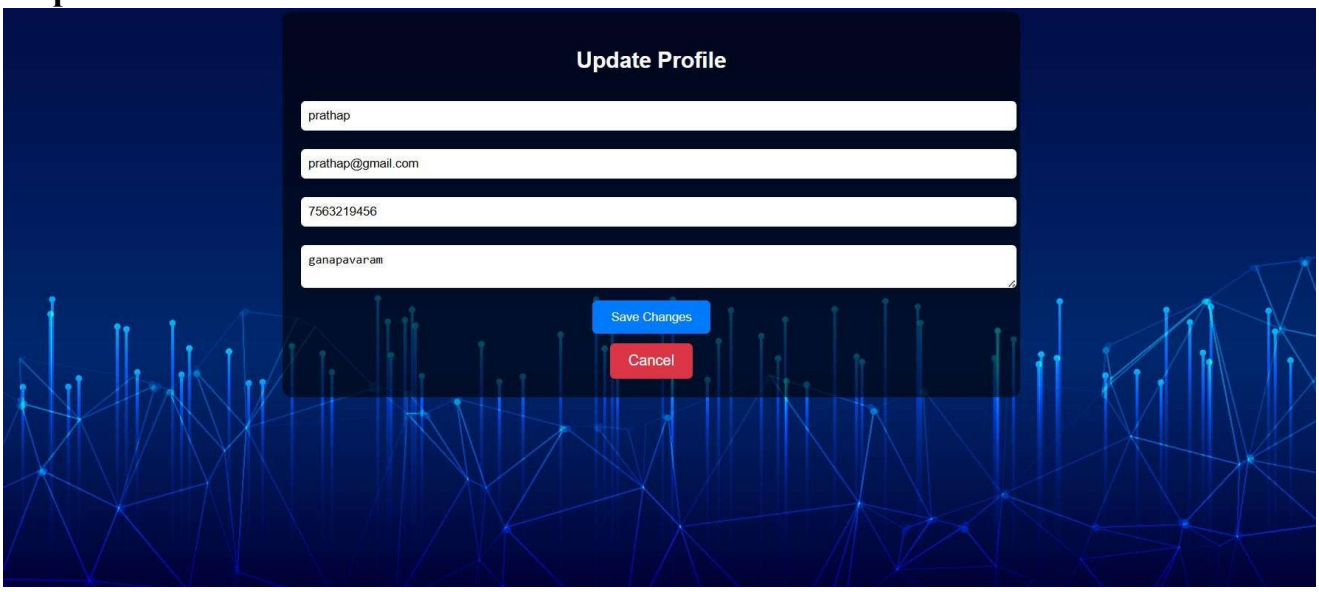
UpdateProfile.jsp:

```
<%@ page import="java.sql.*"%>
<%
HttpSession userSession = request.getSession(false);
if (userSession == null || userSession.getAttribute("account_number") == null) {
    response.sendRedirect("login.jsp");
    return;
}

Connection con = null;
PreparedStatement ps = null;
ResultSet rs = null;

try {
    con = dao.DatabaseConnection.getConnection();
    String accNumberStr = (String) userSession.getAttribute("account_number");
    long accNumber = Long.parseLong(accNumberStr);

    String query = "SELECT name, email, contact, address FROM users WHERE account_number=?";
    ps = con.prepareStatement(query);
    ps.setLong(1, accNumber);
    rs = ps.executeQuery();

    String name = "", email = "", contact = "", address = "";
    if (rs.next()) {
        name = rs.getString("name");
        email = rs.getString("email");
        contact = rs.getString("contact");
        address = rs.getString("address");
    }
%>

<!DOCTYPE html>
<html>
<head>
    <title>Update Profile</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: url('background.jpg') no-repeat center center fixed;
            background-size: cover;
            color: white;
            text-align: center;
        }

        .container {
            width: 50%;
            margin: 50px auto;
            background: rgba(0, 0, 0, 0.6);
            padding: 20px;
            border-radius: 10px;
        }
```

```
            border-radius: 10px;
        }

        input, textarea {
            width: 100%;
            padding: 8px;
            margin: 10px 0;
            border: none;
            border-radius: 5px;
        }

        .update-btn {
            padding: 10px 20px;
            background: #007bff;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }

        .update-btn:hover {
            background: #0056b3;
        }

        .back-btn {
            padding: 10px 20px;
            background: #dc3545;
            color: white;
            text-decoration: none;
            border-radius: 5px;
            display: inline-block;
            margin-top: 10px;
        }

        .back-btn:hover {
            background: #c82333;
        }
    </style>
</head>
<body>

    <div class="container">
        <h2>Update Profile</h2>
        <form action="UpdateProfileServlet" method="post">
            <input type="text" name="name" value="<%= name %>" required>
            <input type="email" name="email" value="<%= email %>" required>
            <input type="text" name="contact" value="<%= contact %>" required>
            <textarea name="address" required><%= address %></textarea>
            <button type="submit" class="update-btn">Save Changes</button>
        </form>
        <a href="profile.jsp" class="back-btn">Cancel</a>
```

```
        </form>
        <a href="profile.jsp" class="back-btn">Cancel</a>
    </div>

</body>
</html>

<%
} catch (Exception e) {
    e.printStackTrace();
    out.println("<p style='color:red;'>Error fetching profile details.</p>");
} finally {
    if (rs != null) try { rs.close(); } catch (SQLException e) {}
    if (ps != null) try { ps.close(); } catch (SQLException e) {}
    if (con != null) try { con.close(); } catch (SQLException e) {}
}
%>
```

Output:



Password hashing:

```
use BankingManagement;

select * from users;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

user_id	name	address	contact	email	password	account_number
7	prathap	ganapavaram	7563219456	prathap@gmail.com	$2a$10$dHz//KkPNlWL.1rYNi/KC.IJEqYFyOAeTk...	1321415460648
8	srikanth	GUNTUR	9876054321	sri123@gmail.com	$2a$10$60xHp63khm4d7TqLs78AquGQlheKv/r...	9610714919431
NULL	NULL	NULL	NULL	NULL	NULL	NULL

10. Conclusion

The Banking Information System successfully provides a secure, efficient, and user-friendly platform for managing banking transactions. By implementing secure authentication, robust transaction processing, and structured data management, the system ensures reliability and scalability. Future enhancements will further improve security and usability, making the system more comprehensive for real-world applications.

Project Developed By: Syamala Prathap Reddy
Development Duration: 6 Weeks
Tools Used: Eclipse, Apache Tomcat, MySQL, Java, JSP, Servlets