Project Report: Registration & Login System

Course: Database Management Systems Lab (CSE 3522)
Section: G | Semester: Fall – 2025
Student Name: Samia Hawlader
Student ID: 011 233 1060
Date: January 04, 2026

1. Introduction

For this assignment, I have created a Registration and Login System using Java and MySQL database. The main goal was to implement a working system where users can register, login, and see their data on a dashboard. I used Java Swing for the user interface and JDBC for connecting to the MySQL database.

The system includes password encryption for security and shows database query results after login, which was the main requirement of this project.

2. Technologies Used

- **Programming Language:** Java (JDK 8+)
- **User Interface:** Java Swing
- **Database:** MySQL
- **Database Driver:** mysql-connector-java.jar
- **Password Security:** SHA-256 hashing algorithm

I chose Java Swing because I am familiar with it from previous courses, and MySQL because it's easy to set up and use.

3. Database Design

Database Name: user_management_system

Table Name: users

The users table has the following columns:
- id (INT, Primary Key, Auto Increment)
- name (VARCHAR 100)
- email (VARCHAR 100, UNIQUE)
- password (VARCHAR 200) - stores encrypted password
- phone (VARCHAR 15)
- gender (VARCHAR 10)
- registration_date (TIMESTAMP, default current time)

I made sure the email field is unique so that no two users can register with the same email address.

4. Implementation Details

4.1 Registration Page

In the registration module, I implemented the following features:

- Input fields for name, email, password, phone number, and gender

- Validation to check if all fields are filled
- Email format validation (must contain @)
- Phone number validation (minimum 10 digits)
- Checking if email already exists in database before registration
- Password encryption using SHA-256 before storing in database

Validation Rules I Used:
- Name cannot be empty
- Email must contain @ symbol and must be unique
- Password must not be empty (minimum 6 characters would be better, but I kept it simple)
- Phone must have at least 10 digits
- Gender is selected from dropdown

4.2 Login Page

The login module includes:

- Email and password input fields
- Database query to check if user exists
- Password verification by comparing encrypted passwords
- Error messages for wrong email or password
- Redirect to dashboard after successful login

I used PreparedStatement to prevent SQL injection attacks, which we learned about in class.

4.3 Dashboard Page

After successful login, the dashboard shows:

- Welcome message with the logged-in user's name
- User's email address
- A table showing all registered users from the database
- Total count of registered users
- Logout button to go back to login page

The dashboard loads data from the database using a SELECT query and displays it in a JTable.

5. Security Features

I implemented several security features in this project:

Password Encryption: I used SHA-256 hashing algorithm to encrypt passwords before storing them in the database. This means even if someone accesses the database, they cannot see the actual passwords.

SQL Injection Prevention: I used PreparedStatement instead of regular Statement for all database queries. This prevents SQL injection attacks.

Email Uniqueness: The database has a UNIQUE constraint on the email field, so duplicate emails are automatically rejected.

Session Management: When a user logs in successfully, their information is passed to the dashboard, and they can logout when done.

6. Testing

I tested the system with different scenarios:

Test Case	What I Tested	Result
TC-01	Register with valid information	Works correctly
TC-02	Try to register with duplicate email	Shows error message
TC-03	Login with correct email and password	Successfully logged in
TC-04	Login with wrong password	Shows error message
TC-05	Dashboard displays all users from database	Data loads properly
TC-06	Logout button	Returns to login page

All test cases passed successfully.

7. Challenges I Faced

While working on this project, I faced a few challenges:

1. **Password Hashing:** At first, I didn't know how to implement password encryption. I researched online and found that Java's MessageDigest class can be used for SHA-256 hashing.

2. **JDBC Connection:** Initially, I had trouble connecting to the database because I forgot to add the MySQL connector JAR file to my project. After adding it to the classpath, it worked fine.

3. **JTable Display:** Loading data into the JTable was a bit tricky. I had to learn about DefaultTableModel to add rows dynamically.

8. What I Learned

From this project, I learned:
- How to connect Java applications with MySQL database using JDBC
- How to use PreparedStatement for secure database queries
- How to implement password hashing for security

- How to work with JTable to display database records
- The importance of input validation

9. Conclusion

I successfully completed all the requirements of this assignment. The system can register new users, verify login credentials, and display database information on the dashboard. I implemented password encryption and prevented SQL injection for security.

The project helped me understand how real-world login systems work and how databases are connected to applications. Overall, I am satisfied with the result and I learned a lot about database connectivity and security.

Screenshots are attached separately.