

**A JAVA MINI PROJECT REPORT**

|  |  |  |
| --- | --- | --- |
| **Submitted** | **by** |  |

**C S VISHNUPRIYA 231501185**

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM CHENNAI-602105 2024 - 2025

ABSTRACT

The Login Page Application is a Java-based project designed to provide secure and user-friendly access control for software systems. This project implements a graphical user interface (GUI) using tools like NetBeans, enabling users to log in or sign up with unique credentials. The application ensures security by incorporating data validation, password encryption, and secure database connectivity.

The project connects to a backend database, such as MySQL, to manage user data, including registration details, login credentials, and user roles. It leverages JDBC for seamless communication between the application and the database. The signup functionality allows new users to create accounts, while the login functionality validates user credentials against the stored data. Advanced features such as password recovery and role-based access can also be included to enhance the system's usability and scalability.

This project demonstrates fundamental Java programming concepts, including GUI development, database interaction, and exception handling. It is designed as a foundational module that can be integrated into larger systems, such as e-commerce platforms, employee management systems, or educational portals. By implementing this project, developers gain hands-on experience in developing secure and efficient access control mechanisms for modern software applications.

INTRODUCTION

The Login System is a desktop application developed to manage secure access for users in various software systems, providing a robust, efficient, and user-friendly solution for authentication. This project enables users to log in with their unique credentials and allows new users to register securely, while administrators can manage user accounts and monitor access. Unlike traditional manual login systems, this software offers a streamlined, automated approach to user authentication and account management.

Users can create accounts with a unique username and password during the signup process, and the system ensures that credentials are securely stored and validated. The login functionality allows users to access their accounts after verifying their details against the database. Administrators can manage user records and enforce security protocols, ensuring the integrity of user data and restricting unauthorized access.

The application employs MySQL as the backend database for reliable storage of user information and integrates Java for the front-end interface. It demonstrates the principles of database connectivity using JDBC and ensures data security with features like password encryption. This project highlights key concepts in software development, offering a secure and intuitive platform for managing user authentication processes.

OBJECTIVES

Primary Objectives

1. **Develop a Secure and User-Friendly Login System**: Create an intuitive interface that allows users to easily log in or register, ensuring a smooth and hassle-free authentication process.
2. **Enhance Data Security**: Implement measures to secure user credentials through password encryption and validation, ensuring the confidentiality and integrity of user data.
3. **Enable Account Management for Administrators**: Provide administrators with tools to manage user accounts, enforce policies, and monitor system access effectively.

Technical Objectives

1. **Reliable Database Management**: Design a robust MySQL database to store and manage user credentials and account details securely.
2. **Seamless Integration of Java and MySQL**: Ensure smooth communication between the Java-based front-end and MySQL backend using JDBC for efficient CRUD operations.
3. **Error Handling and Validation**: Incorporate comprehensive error handling and data validation techniques to minimize runtime errors and prevent invalid inputs.

Business Objectives

1. **Promote Secure Access Control**: Equip organizations with a reliable system to manage user authentication and prevent unauthorized access.
2. **Increase User Satisfaction**: Deliver an intuitive and efficient login experience to improve user engagement and satisfaction with the system.
3. **Strengthen Trust with Robust Security**: Implement secure data handling practices to protect sensitive user information, building trust in the application.

**SOFTWARE DESCRIPTION**

**JAVA**

Java is a powerful, object-oriented programming language widely used for developing secure and scalable applications. Known for its portability and platform independence, Java provides the foundation for the Login System. It is used to implement core functionalities such as user registration, login authentication, and error handling.

Java’s robust library ecosystem simplifies the development of graphical user interfaces (GUIs) and backend logic. In this project, Java serves as the primary programming language, handling processes like input validation, password encryption, and data interaction with the database. Its versatility ensures the application can run seamlessly across multiple platforms.

**MYSQL**

MySQL is a reliable and efficient open-source relational database management system (RDBMS) used for storing and managing user data. It organizes information in structured tables, enabling secure storage and retrieval of sensitive login credentials. MySQL's compatibility with Java makes it a natural choice for the backend of the Login System.

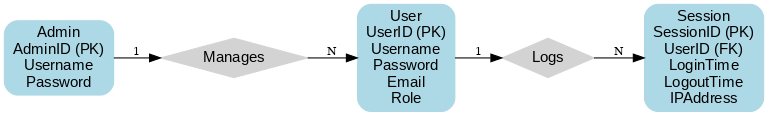
In this project, MySQL is used to store user information, including usernames, encrypted passwords, and account details. SQL queries are utilized for operations like user authentication, data insertion, and account management. MySQL ensures data integrity and security, providing a solid foundation for the application's database layer.

**JDBC (Java Database Connectivity)**

JDBC is a Java API that facilitates communication between Java applications and relational databases like MySQL. It allows for the execution of SQL queries, retrieval of data, and seamless integration of the application with the database.

In this project, JDBC is used to connect the Java-based Login System with the MySQL database. It handles tasks such as validating user credentials during login, storing encrypted passwords during registration, and executing queries to retrieve or update user information. By leveraging JDBC, the project ensures efficient and secure database interactions.

ER DIAGRAM



DATABASE CODE

-- Create the database

CREATE DATABASE LoginSystem;

-- Use the database

USE LoginSystem;

-- Create the Users table

CREATE TABLE Users (

UserID INT AUTO\_INCREMENT PRIMARY KEY, -- Unique identifier for the user

Username VARCHAR(50) NOT NULL UNIQUE, -- Username (must be unique)

PasswordHash VARCHAR(255) NOT NULL -- Encrypted password

);

SOURCE CODE

// LOGIN PAGE

package javaproject1;

import javax.swing.JFrame;

public class Login extends javax.swing.JFrame {

public Login() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton2 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

Right = new javax.swing.JPanel();

jLabel5 = new javax.swing.JLabel();

Left = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jPasswordField1 = new javax.swing.JPasswordField();

jButton1 = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jButton2.setBackground(new java.awt.Color(0, 153, 153));

jButton2.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jButton2.setText("Sign Up");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("LOGIN");

setBackground(new java.awt.Color(204, 204, 204));

setPreferredSize(new java.awt.Dimension(600, 500));

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel1.setPreferredSize(new java.awt.Dimension(600, 500));

jPanel1.setLayout(null);

Right.setBackground(new java.awt.Color(0, 153, 153));

Right.setPreferredSize(new java.awt.Dimension(400, 500));

jLabel5.setIcon(new javax.swing.ImageIcon("C:\\Users\\priya\\OneDrive\\Documents\\NetBeansProjects\\Javaproject1\\src\\icon\\logo.jpg")); // NOI18N

javax.swing.GroupLayout RightLayout = new javax.swing.GroupLayout(Right);

Right.setLayout(RightLayout);

RightLayout.setHorizontalGroup(

RightLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, RightLayout.createSequentialGroup()

.addContainerGap(75, Short.MAX\_VALUE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 249, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(46, 46, 46))

);

RightLayout.setVerticalGroup(

RightLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, RightLayout.createSequentialGroup()

.addContainerGap(104, Short.MAX\_VALUE)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(73, 73, 73))

);

jPanel1.add(Right);

Right.setBounds(0, 0, 370, 400);

Left.setBackground(new java.awt.Color(255, 255, 255));

jLabel1.setFont(new java.awt.Font("Times New Roman", 3, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(0, 102, 102));

jLabel1.setText("LOGIN");

jLabel2.setBackground(new java.awt.Color(0, 102, 102));

jLabel2.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel2.setForeground(new java.awt.Color(0, 153, 153));

jLabel2.setText("Email:");

jTextField1.setForeground(new java.awt.Color(0, 102, 102));

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jLabel3.setBackground(new java.awt.Color(0, 102, 102));

jLabel3.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jLabel3.setForeground(new java.awt.Color(0, 153, 153));

jLabel3.setText("Password:");

jButton1.setBackground(new java.awt.Color(0, 153, 153));

jButton1.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

jButton1.setText("LOGIN");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Times New Roman", 0, 10)); // NOI18N

jLabel4.setText("I don't have an account");

javax.swing.GroupLayout LeftLayout = new javax.swing.GroupLayout(Left);

Left.setLayout(LeftLayout);

LeftLayout.setHorizontalGroup(

LeftLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(LeftLayout.createSequentialGroup()

.addGap(21, 21, 21)

.addGroup(LeftLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 83, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 175, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 59, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(LeftLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 175, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4))

.addContainerGap(34, Short.MAX\_VALUE))

);

LeftLayout.setVerticalGroup(

LeftLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(LeftLayout.createSequentialGroup()

.addGap(105, 105, 105)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(25, 25, 25)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(81, Short.MAX\_VALUE))

);

jPanel1.add(Left);

Left.setBounds(370, 0, 230, 400);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 732, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 401, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 105, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

SignUp SignUpFrame = new SignUp(); // Use a more descriptive name

SignUpFrame.setVisible(true);

SignUpFrame.pack();

SignUpFrame.setLocationRelativeTo(null);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

JFrame loggedInFrame = new JFrame("Logged In");

loggedInFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

LoggedIn loggedInPage = new LoggedIn();

loggedInFrame.add(loggedInPage);

loggedInFrame.pack();

loggedInFrame.setLocationRelativeTo(null);

loggedInFrame.setVisible(true);

this.dispose();

}

// Variables declaration - do not modify

private javax.swing.JPanel Left;

private javax.swing.JPanel Right;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

// LOGGED IN PAGE

package javaproject1;

public class LoggedIn extends javax.swing.JPanel {

public LoggedIn() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jPanel1.setBackground(new java.awt.Color(0, 204, 204));

jPanel2.setBackground(new java.awt.Color(255, 255, 255));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 231, Short.MAX\_VALUE)

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 463, Short.MAX\_VALUE)

);

jLabel2.setIcon(new javax.swing.ImageIcon("C:\\Users\\priya\\OneDrive\\Documents\\NetBeansProjects\\Javaproject1\\src\\icon\\logo.jpg")); // NOI18N

jLabel1.setFont(new java.awt.Font("Times New Roman", 0, 24)); // NOI18N

jLabel1.setText("LOGGED IN!");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(304, 304, 304)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 307, Short.MAX\_VALUE)))

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel2)

.addGap(18, 18, 18)

.addComponent(jLabel1)

.addGap(56, 56, 56))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 42, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

}// </editor-fold>

// Variables declaration - do not modify

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

// End of variables declaration

void setLocationRelativeTo(Object object) {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

void pack() {

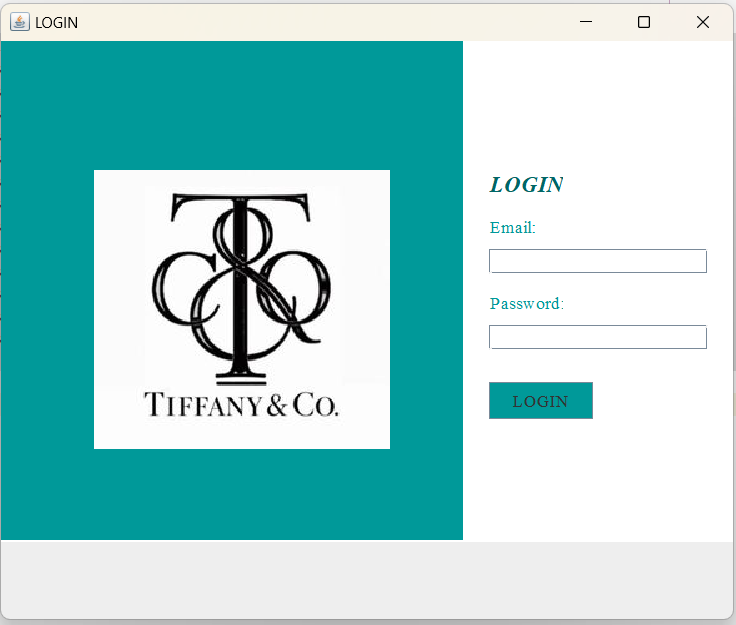
throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

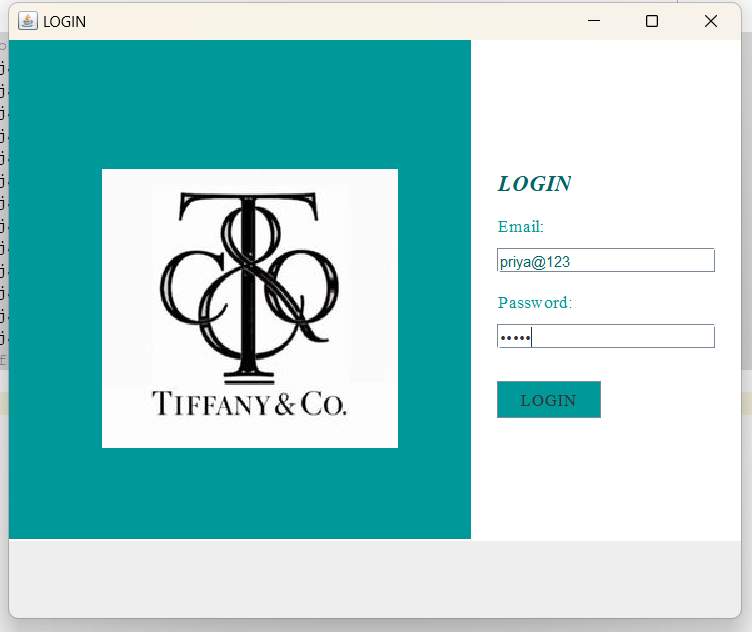
}

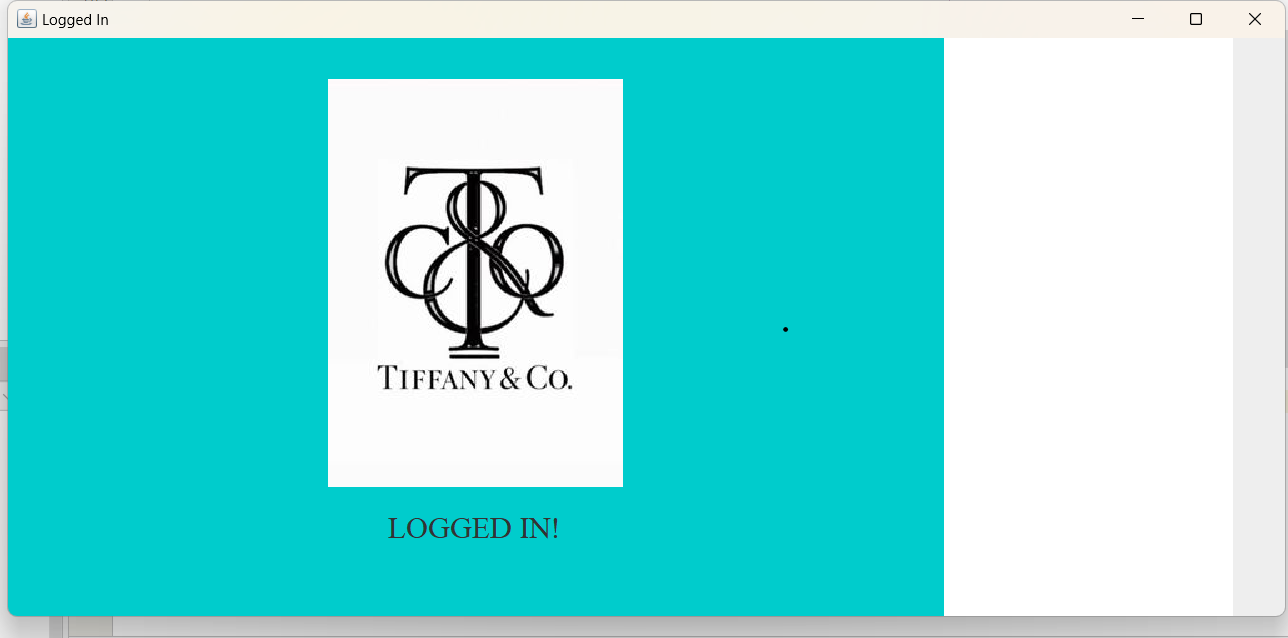
}

RESULTS

LOGIN IN PAGE:







OVERALL SUMMARY

The Login System project successfully achieves its core objectives by providing a secure and user-friendly interface for user authentication and account management. With essential features like user registration, login validation, and password encryption, the system ensures that users can access their accounts safely and efficiently. The project also facilitates seamless integration with a MySQL database, enabling reliable storage and retrieval of user credentials.

While the current version focuses on the foundational functionalities of a login system, additional features such as password recovery, multi-factor authentication, or role-based access control could be incorporated in future updates to further enhance security and user experience. The system's robust architecture and modular design provide a strong foundation for scalability, ensuring that it can adapt to the growing needs of users and organizations.

CONCLUSION

The Login System project has successfully met its primary objectives, offering a secure and efficient platform for user authentication. It provides a seamless experience for users to register, log in, and access their accounts while ensuring their credentials are securely managed through password encryption and database integration. The MySQL database, managed through SQL Workbench, forms a reliable backbone for storing user data, and the Java-based implementation ensures a smooth and user-friendly interface.

From a user perspective, the system offers an intuitive interface that simplifies the login and registration processes. Key features like unique user credentials and password validation ensure that only authorized users can access their accounts, fulfilling the core purpose of the project.

From an administrative perspective, while the current system does not include a dedicated admin panel, the robust database design supports potential enhancements such as user management, role-based access control, and activity tracking. These features could be added in future iterations to improve oversight and administrative efficiency.

These features would strengthen the system’s defenses against unauthorized access and provide users with more flexible account recovery options.

The use of SQL Workbench as the database management tool facilitated efficient database design, query execution, and troubleshooting, ensuring smooth integration between the front-end and back-end components of the project.

Overall, the Login System lays a strong foundation for future enhancements. With opportunities for feature expansion and security upgrades, this system is well-positioned to evolve into a scalable and secure solution for managing user authentication in various applications, from small-scale systems to enterprise-level platforms.

REFERENCES

Java and MySQL Integration:

* **MySQL Documentation:**  
  Provides in-depth explanations and guidelines for connecting Java applications to MySQL, which is essential for your project's database integration.  
  Available at: [MySQL Documentation](https://dev.mysql.com/doc)

Java Programming:

* **Oracle Java Documentation:**  
  Official Java documentation to understand Java programming concepts and APIs, particularly relevant for building the backend of your voting system.  
  Available at: [Oracle Java Documentation](https://docs.oracle.com/en/java/)