Mini Project – Train Booking System

AIM:

To Create a java swing project on train booking system with login and registration with database connection.

DatabaseConnection.java (used to make connection to MYSQL database with mysql\_connector\_9.0.1.jar file)

CODE:

import javax.swing.\*; import java.awt.\*; import java.awt.event.\*; import java.sql.\*; import java.util.List;

import java.util.ArrayList;

public class TrainBookingSystem extends JFrame {

private Connection connection;

public TrainBookingSystem() { // Initialize the database connection try {

connection = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/train\_booking\_system", "root",

"Udaya@2006");

} catch (SQLException e) { e.printStackTrace();

JOptionPane.showMessageDialog(this, "Database connection failed!"); System.exit(1);

}

// Show the login form initially showLoginForm();

}

// Method to show Login Form private void showLoginForm() {

JPanel panel = new JPanel(new GridBagLayout()); GridBagConstraints gbc = new GridBagConstraints(); gbc.insets = new Insets(5, 5, 5, 5);

// Set background image for the login page

JLabel background = new JLabel(new ImageIcon("login\_background.jpg")); background.setLayout(new BorderLayout());

JLabel titleLabel = new JLabel("Train Booking System - Login", SwingConstants.CENTER);

titleLabel.setFont(new Font("Arial", Font.BOLD, 20)); titleLabel.setForeground(Color.WHITE);

JTextField usernameField = new JTextField(15);

JPasswordField passwordField = new JPasswordField(15);

JButton loginButton = new JButton("Login");

JButton signupButton = new JButton("Sign Up");

// Set font and styling for fields and buttons usernameField.setFont(new Font("Arial", Font.PLAIN, 14)); passwordField.setFont(new Font("Arial", Font.PLAIN, 14)); loginButton.setFont(new Font("Arial", Font.BOLD, 14)); signupButton.setFont(new Font("Arial", Font.BOLD, 14));

gbc.gridx = 0; gbc.gridy = 0;

gbc.gridwidth = 2;

background.add(titleLabel, BorderLayout.NORTH);

gbc.gridwidth = 1;

gbc.gridy++;

panel.add(new JLabel("Username:"), gbc); gbc.gridx = 1;

panel.add(usernameField, gbc);

gbc.gridx = 0; gbc.gridy++;

panel.add(new JLabel("Password:"), gbc);

gbc.gridx = 1;

panel.add(passwordField, gbc);

gbc.gridx = 0; gbc.gridy++;

panel.add(loginButton, gbc);

gbc.gridx = 1;

panel.add(signupButton, gbc);

loginButton.addActionListener(e -> {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (authenticateUser(username, password)) {

JOptionPane.showMessageDialog(this, "Login Successful!"); showTrainSelectionForm(); // Go to train selection after login

} else {

JOptionPane.showMessageDialog(this, "Invalid Credentials!");

}

});

signupButton.addActionListener(e -> showSignupForm());

background.add(panel, BorderLayout.CENTER); setContentPane(background);

setSize(600, 400); setLocationRelativeTo(null);

setDefaultCloseOperation(EXIT\_ON\_CLOSE); setVisible(true);

}

// Method to show the Signup Form private void showSignupForm() {

JPanel panel = new JPanel(new GridBagLayout()); GridBagConstraints gbc = new GridBagConstraints(); gbc.insets = new Insets(5, 5, 5, 5);

// Set background image for signup page

JLabel background = new JLabel(new ImageIcon("signup\_background.jpg")); background.setLayout(new BorderLayout());

JLabel titleLabel = new JLabel("Train Booking System - Sign Up", SwingConstants.CENTER);

titleLabel.setFont(new Font("Arial", Font.BOLD, 20)); titleLabel.setForeground(Color.WHITE);

JTextField usernameField = new JTextField(15);

JPasswordField passwordField = new JPasswordField(15);

JTextField emailField = new JTextField(15);

JButton registerButton = new JButton("Register");

JButton backButton = new JButton("Back");

// Set font and styling for fields and buttons usernameField.setFont(new Font("Arial", Font.PLAIN, 14)); passwordField.setFont(new Font("Arial", Font.PLAIN, 14)); emailField.setFont(new Font("Arial", Font.PLAIN, 14)); registerButton.setFont(new Font("Arial", Font.BOLD, 14)); backButton.setFont(new Font("Arial", Font.BOLD, 14));

gbc.gridx = 0; gbc.gridy = 0;

gbc.gridwidth = 2;

background.add(titleLabel, BorderLayout.NORTH);

gbc.gridwidth = 1;

gbc.gridy++;

panel.add(new JLabel("Username:"), gbc); gbc.gridx = 1;

panel.add(usernameField, gbc);

gbc.gridx = 0; gbc.gridy++;

panel.add(new JLabel("Password:"), gbc); gbc.gridx = 1;

panel.add(passwordField, gbc);

gbc.gridx = 0; gbc.gridy++;

panel.add(new JLabel("Email:"), gbc);

gbc.gridx = 1;

panel.add(emailField, gbc);

gbc.gridx = 0; gbc.gridy++;

panel.add(registerButton, gbc);

gbc.gridx = 1;

panel.add(backButton, gbc); registerButton.addActionListener(e -> {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

String email = emailField.getText();

if (registerUser(username, password, email)) {

JOptionPane.showMessageDialog(this, "Registration Successful!"); showLoginForm();

} else {

JOptionPane.showMessageDialog(this, "Registration Failed!");

}

});

backButton.addActionListener(e -> showLoginForm());

background.add(panel, BorderLayout.CENTER); setContentPane(background); revalidate();

}

// Method to show the Train Selection Form (Ticket Booking)

private void showTrainSelectionForm() {

JPanel panel = new JPanel(new BorderLayout());

JLabel titleLabel = new JLabel("Train Selection", SwingConstants.CENTER); titleLabel.setFont(new Font("Arial", Font.BOLD, 20));

// Query to fetch available trains (you should have this table in your database) String query = "SELECT \* FROM trains";

try (Statement stmt = connection.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

String[] columnNames = {"Train ID", "Train Name", "Departure", "Arrival", "Fare"};

Object[][] data = new Object[10][5]; // Assuming 10 trains for now

int i = 0;

while (rs.next() && i < 10) { data[i][0] = rs.getInt("train\_id"); data[i][1] = rs.getString("train\_name"); data[i][2] = rs.getString("departure"); data[i][3] = rs.getString("arrival");

data[i][4] = rs.getDouble("fare"); i++;

}

JTable table = new JTable(data, columnNames); JScrollPane scrollPane = new JScrollPane(table); panel.add(scrollPane, BorderLayout.CENTER);

// Add button to book tickets

JButton bookButton = new JButton("Book Ticket"); panel.add(bookButton, BorderLayout.SOUTH);

bookButton.addActionListener(e -> { int selectedRow = table.getSelectedRow();

if (selectedRow != -1) {

// Get the train details from the selected row int trainId = (int) table.getValueAt(selectedRow, 0);

String trainName = (String) table.getValueAt(selectedRow, 1); double fare = (double) table.getValueAt(selectedRow, 4);

// Now you can show a ticket booking form or proceed with booking

JOptionPane.showMessageDialog(this,

"You selected train: " + trainName + "\nFare: " + fare); } else {

JOptionPane.showMessageDialog(this, "Please select a train.");

}

});

} catch (SQLException e) { e.printStackTrace();

JOptionPane.showMessageDialog(this, "Failed to load train data.");

}

setContentPane(panel);

revalidate();

}

// Authentication logic

private boolean authenticateUser(String username, String password) {

String query = "SELECT \* FROM users WHERE username = ? AND password =

?";

try (PreparedStatement stmt = connection.prepareStatement(query)) { stmt.setString(1, username); stmt.setString(2, password);

ResultSet rs = stmt.executeQuery(); return rs.next();

} catch (SQLException e) { e.printStackTrace();

return false;

}

}

// Registration logic

private boolean registerUser(String username, String password, String email) {

String query = "INSERT INTO users (username, password, email) VALUES (?, ?,

?)";

try (PreparedStatement stmt = connection.prepareStatement(query)) { stmt.setString(1, username); stmt.setString(2, password);

stmt.setString(3, email);

stmt.executeUpdate();

return true;

} catch (SQLException e) { e.printStackTrace(); return false;

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(TrainBookingSystem::new);

} }

OUTPUT:











