

(Department of Computer Science)

(OOP Final Semester Project Task )

Group Members

Saad Akbar

Arslan Aslam

## Admin Login:

|  |
| --- |
| import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.\*;  public class AdminLogin extends JFrame implements ActionListener{  private JLabel label;  private JButton adminlogin, librarianlogin, exit;  public AdminLogin(){  setTitle("Library Management System");  setSize(420, 300);  setResizable(false);  setLocation(430,150);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setLayout(null);      label = new JLabel("Library Management System");  label.setBounds(100,30,250,50);  label.setFont(new Font("Garamond", Font.BOLD, 17));  label.setForeground(Color.black);  add(label);    //setting buttons  String code = "#F5FFFA";  adminlogin = new JButton("Admin-Login");  adminlogin.setBounds(152, 85, 110, 25);  adminlogin.setFont(new Font("Arial", Font.BOLD, 10));  adminlogin.setBackground(Color.decode(code));  adminlogin.setFocusable(false);  adminlogin.addActionListener(this);  add(adminlogin);      librarianlogin = new JButton("Librarian Login");  librarianlogin.setBounds(152, 130, 110, 25);  librarianlogin.setFont(new Font("Arial", Font.BOLD, 10));  librarianlogin.setBackground(Color.decode(code));  librarianlogin.setFocusable(false);  librarianlogin.addActionListener(this);  add(librarianlogin);    exit = new JButton("EXIT");  exit.setBounds(152, 175, 110, 25);  exit.setFont(new Font("Arial", Font.BOLD, 10));  exit.setBackground(Color.decode(code));  exit.setFocusable(false);  exit.addActionListener(this);  add(exit);    ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420-300.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 300);    add(l2);    setVisible(true);  }    public static void main(String[] args) {  // TODO Auto-generated method stub  new AdminLogin();  }  @Override  public void actionPerformed(ActionEvent e) {  if(e.getSource() == exit)  {  System.exit(0);  } else if(e.getSource() == adminlogin)  {  setVisible(false);  new adminform();  } else if(e.getSource() == librarianlogin) {  setVisible(false);  new librarianlogin();  }    }  } Admin Form: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.\*;  public class adminform extends JFrame implements ActionListener {  private JLabel title, usernameLabel, passwordLabel;  private JTextField usernameField;  private JPasswordField passwordField;  private JButton loginButton;  public adminform() {  setSize(420, 300);  setResizable(false);  setLayout(null);  title = new JLabel("Admin Login Form");  title.setBounds(120, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.black);  usernameLabel = new JLabel("Enter Name:");  usernameLabel.setBounds(50, 100, 150, 20);  usernameLabel.setFont(new Font("Garamond", Font.BOLD, 15));  usernameLabel.setForeground(Color.black);  usernameField = new JTextField();  usernameField.setBounds(200, 100, 150, 20);    passwordLabel = new JLabel("Enter Password:");  passwordLabel.setBounds(50, 140, 200, 20);  passwordLabel.setFont(new Font("Garamond", Font.BOLD, 15));  passwordLabel.setForeground(Color.black);  passwordField = new JPasswordField();  passwordField.setBounds(200, 140, 150, 20);    String code = "#F5FFFA";    loginButton = new JButton("Login");  loginButton.setBounds(160, 200, 100, 30);  loginButton.setFont(new Font("Arial", Font.BOLD, 10));  loginButton.setBackground(Color.decode(code));  loginButton.setFocusable(false);  loginButton.addActionListener(this);  add(title);  add(usernameLabel);  add(usernameField);  add(passwordLabel);  add(passwordField);  add(loginButton);    ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420-300.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 300);    add(l2);    setVisible(true);  }  public static void main(String[] args) {  new adminform();  }  @Override  public void actionPerformed(ActionEvent e) {  String user = usernameField.getText();  String password = passwordField.getText();  if(e.getSource() == loginButton)    {  if(user.equals("admin") && password.equals("admin123")) {  JOptionPane.showMessageDialog(null,"Login Successfully.","Message",JOptionPane.INFORMATION\_MESSAGE);  setVisible(false);  new adminSection();  }  else  {  JOptionPane.showMessageDialog(null,"Please make sure you entered correct data.","Alert",JOptionPane.ERROR\_MESSAGE);  }    }    }  } Admin Section: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.\*;  public class adminSection extends JFrame implements ActionListener {  private JLabel label;  private JButton addLibrarian, viewLibrarian, deleteLibrarian, logout;  public adminSection() {  setTitle("Admin Section");  setSize(420, 300);  setResizable(false);  setLocation(430, 150);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  label = new JLabel("Admin Section");  label.setBounds(152, 15, 250, 50);  label.setFont(new Font("Garamond", Font.BOLD, 17));  label.setForeground(Color.black);  // Setting buttons  String code = "#F5FFFA";    addLibrarian = new JButton("Add Librarian");  addLibrarian.setBounds(152, 85, 110, 25);  addLibrarian.setFont(new Font("Arial", Font.BOLD, 10));  addLibrarian.setBackground(Color.decode(code));  addLibrarian.setFocusable(false);  addLibrarian.addActionListener(this);  add(addLibrarian);  viewLibrarian = new JButton("View Librarian");  viewLibrarian.setBounds(152, 130, 110, 25);  viewLibrarian.setFont(new Font("Arial", Font.BOLD, 10));  viewLibrarian.setBackground(Color.decode(code));  viewLibrarian.setFocusable(false);  viewLibrarian.addActionListener(this);  add(viewLibrarian);    deleteLibrarian = new JButton("Delete Librarian");  deleteLibrarian.setBounds(152, 175, 110, 25);  deleteLibrarian.setFont(new Font("Arial", Font.BOLD, 10));  deleteLibrarian.setBackground(Color.decode(code));  deleteLibrarian.setFocusable(false);  deleteLibrarian.addActionListener(this);  add(deleteLibrarian);  logout = new JButton("Logout");  logout.setBounds(152, 220, 110, 25);  logout.setFont(new Font("Arial", Font.BOLD, 10));  logout.setBackground(Color.decode(code));  logout.setFocusable(false);  logout.addActionListener(this);  add(logout);  setLayout(null);  add(label);    ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420-300.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 300);    add(l2);    setVisible(true);  }  public static void main(String[] args) {  new adminSection();  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == logout) {  setVisible(false);  new AdminLogin();  } else if (e.getSource() == addLibrarian) {  // Perform actions for adding a librarian  setVisible(false);  new addlibrarian();  } else if (e.getSource() == viewLibrarian) {  // Perform actions for viewing librarians  setVisible(false);  new viewLibrarian();  } else if (e.getSource() == deleteLibrarian) {  // Perform actions for viewing librarians  setVisible(false);  new deletelibrarian();  }  }  } Add Librarian: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import javax.swing.\*;  public class addlibrarian extends JFrame implements ActionListener {  private JLabel title, username, passwordlabel, email, address, city, contactNo;  private JTextField usernameField, emailField, addressField, cityField, contactNoField;  private JPasswordField passwordField;  private JButton addLibrarian, back;  public addlibrarian() {  setSize(420, 450);  setResizable(false);  setLayout(null);  title = new JLabel("Add Librarian");  title.setBounds(120, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  username = new JLabel("Name:");  username.setBounds(50, 100, 200, 20);  username.setFont(new Font("Garamond", Font.BOLD, 15));  username.setForeground(Color.BLACK);  passwordlabel = new JLabel("Password:");  passwordlabel.setBounds(50, 130, 200, 20);  passwordlabel.setFont(new Font("Garamond", Font.BOLD, 15));  passwordlabel.setForeground(Color.BLACK);  email = new JLabel("Email:");  email.setBounds(50, 160, 200, 20);  email.setFont(new Font("Garamond", Font.BOLD, 15));  email.setForeground(Color.BLACK);  address = new JLabel("Address:");  address.setBounds(50, 190, 200, 20);  address.setFont(new Font("Garamond", Font.BOLD, 15));  address.setForeground(Color.BLACK);  city = new JLabel("City:");  city.setBounds(50, 220, 200, 20);  city.setFont(new Font("Garamond", Font.BOLD, 15));  city.setForeground(Color.BLACK);  contactNo = new JLabel("Contact No:");  contactNo.setBounds(50, 250, 200, 20);  contactNo.setFont(new Font("Garamond", Font.BOLD, 15));  contactNo.setForeground(Color.BLACK);  usernameField = new JTextField();  usernameField.setBounds(150, 100, 200, 20);  passwordField = new JPasswordField();  passwordField.setBounds(150, 130, 200, 20);  emailField = new JTextField();  emailField.setBounds(150, 160, 200, 20);  addressField = new JTextField();  addressField.setBounds(150, 190, 200, 20);  cityField = new JTextField();  cityField.setBounds(150, 220, 200, 20);  contactNoField = new JTextField();  contactNoField.setBounds(150, 250, 200, 20);    //buttons  String code = "#F5FFFA";  addLibrarian = new JButton("Add Librarian");  addLibrarian.setBounds(255, 320, 120, 30);  addLibrarian.setFont(new Font("Arial", Font.BOLD, 10));  addLibrarian.setBackground(Color.decode(code));  addLibrarian.addActionListener(this);  addLibrarian.setFocusable(false);  back = new JButton("Back");  back.setBounds(30, 320, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.addActionListener(this);  back.setFocusable(false);  add(title);  add(username);  add(passwordlabel);  add(email);  add(address);  add(city);  add(contactNo);  add(usernameField);  add(passwordField);  add(emailField);  add(addressField);  add(cityField);  add(contactNoField);  add(addLibrarian);  add(back);  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420x450.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 450);  add(l2);  setVisible(true);  }  public static void main(String[] args) {  new addlibrarian();  }  @Override  public void actionPerformed(ActionEvent e) {  try {if(e.getSource() == back) {  setVisible(false);  new adminSection();  }  else if (e.getSource() == addLibrarian) {  // Perform the logic to add a librarian  String name = usernameField.getText();  String password1 = passwordField.getText();    String email = emailField.getText();  String address = addressField.getText();  String city = cityField.getText();  String contactNo = contactNoField.getText();    connectSQL connect = new connectSQL();  String sql = "INSERT INTO librarians (name, password, email, address, city, contactNo) VALUES (?, ?, ?, ?, ?, ?)";  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);    statement.setString(1, name);  statement.setString(2, password1);  statement.setString(3, email);  statement.setString(4, address);  statement.setString(5, city);  statement.setString(6, contactNo);  // Execute the query  statement.executeUpdate();    JOptionPane.showMessageDialog(this, "Librarian added successfully!");    usernameField.setText(null);  passwordField.setText(null);  emailField.setText(null);  addressField.setText(null);  cityField.setText(null);  contactNoField.setText(null);      }} catch(Exception E) {  System.out.println(E);  }        }} Delete Librarian: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.SQLException;  import javax.swing.\*;  public class deletelibrarian extends JFrame implements ActionListener {  private JLabel id;  private JTextField userId;  private JButton delete, back;  public deletelibrarian() {  setSize(420, 300);  setResizable(false);  setLayout(null);  id = new JLabel("Enter Id:");  id.setBounds(50, 100, 150, 20);  id.setFont(new Font("Garamond", Font.BOLD, 15));  id.setForeground(Color.black);  userId = new JTextField();  userId.setBounds(200, 100, 150, 20);  //setting buttons  String code = "#F5FFFA";    delete = new JButton("Delete");  delete.setBounds(250, 200, 120, 30);  delete.setFont(new Font("Arial", Font.BOLD, 10));  delete.setBackground(Color.decode(code));  delete.addActionListener(this);  back = new JButton("Back");  back.setBounds(40, 200, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.addActionListener(this);  add(id);  add(userId);  add(delete);  add(back);  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420-300.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 300);  add(l2);  setVisible(true);  }  public static void main(String[] args) {  // TODO Auto-generated method stub  new deletelibrarian();  }  @Override  public void actionPerformed(ActionEvent e) {  try {  if (e.getSource() == back) {  setVisible(false);  new adminSection();  } else if (e.getSource() == delete) {  String userIdText = userId.getText();  connectSQL connect = new connectSQL();  // Delete the librarian from the database based on the provided user ID  String sql = "DELETE FROM librarians WHERE id = ?";  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);  statement.setString(1, userIdText);  int rowsAffected = statement.executeUpdate();  userId.setText("");  if (rowsAffected > 0) {  // Display a message indicating success  JOptionPane.showMessageDialog(this, "Record deleted successfully!");  } else {  // Display a message indicating failure  JOptionPane.showMessageDialog(this, "Failed to delete record!");  }  }  } catch (SQLException ex) {  ex.printStackTrace();  }  }  } View Librarian: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import javax.swing.\*;  import javax.swing.table.DefaultTableModel;  public class viewLibrarian extends JFrame implements ActionListener {  private JTable librarianTable;  private JButton back;  public viewLibrarian() {  setSize(630, 500);  setResizable(false);  setLayout(null);  JLabel title = new JLabel("View Librarian");  title.setBounds(250, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);    //setting buttons  String code = "#808080";    back = new JButton("Back");  back.setBounds(250, 420, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setForeground(Color.white);  back.setFocusable(false);  back.addActionListener(this);  librarianTable = new JTable();  JScrollPane scrollPane = new JScrollPane(librarianTable);  scrollPane.setBounds(30, 100, 550, 300);  add(title);  add(scrollPane);  add(back);      setVisible(true);  loadLibrarianDetails();  }  public static void main(String[] args) {  new viewLibrarian();  }  private void loadLibrarianDetails() {  try {  connectSQL connect = new connectSQL();  String sql = "SELECT \* FROM librarians";  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet rs = statement.executeQuery();  // Create a DefaultTableModel to hold the data  DefaultTableModel model = new DefaultTableModel();  model.addColumn("ID");  model.addColumn("Name");  model.addColumn("Password");  model.addColumn("Email");  model.addColumn("Address");  model.addColumn("City");  model.addColumn("Contact No");  // Populate the table model with the data from the resultSet  while (rs.next()) {    String id = rs.getString("id");  String name = rs.getString("name");  String password = rs.getString("password");  String email = rs.getString("email");  String address = rs.getString("address");  String city = rs.getString("city");  String contactNo = rs.getString("contactNo");  model.addRow(new Object[] { id, name, password, email, address, city, contactNo });  }  rs.close();  statement.close();  connection.close();  // Set the table model to the JTable  librarianTable.setModel(model);  } catch (SQLException ex) {  ex.printStackTrace();  }  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new adminSection();  }  }  } Librarian Login: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import javax.swing.\*;  import javax.swing.JOptionPane;  import javax.swing.JPasswordField;  import javax.swing.JTextField;  public class librarianlogin extends JFrame implements ActionListener {  private JLabel title, usernameLabel, passwordLabel;  private JTextField usernameField;  private JPasswordField passwordField;  private JButton loginButton;  public librarianlogin() {  setSize(420, 300);  setResizable(false);  setLayout(null);  title = new JLabel("Librarian Login Form");  title.setBounds(120, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.black);  usernameLabel = new JLabel("Enter Name:");  usernameLabel.setBounds(50, 100, 150, 20);  usernameLabel.setFont(new Font("Garamond", Font.BOLD, 15));  usernameLabel.setForeground(Color.black);  usernameField = new JTextField();  usernameField.setBounds(200, 100, 150, 20);  passwordLabel = new JLabel("Enter Password:");  passwordLabel.setBounds(50, 140, 200, 20);  passwordLabel.setFont(new Font("Garamond", Font.BOLD, 15));  passwordLabel.setForeground(Color.black);  passwordField = new JPasswordField();  passwordField.setBounds(200, 140, 150, 20);  // setting button  String code = "#F5FFFA";    loginButton = new JButton("Login");  loginButton.setBounds(160, 200, 100, 30);  loginButton.setFont(new Font("Arial", Font.BOLD, 10));  loginButton.setBackground(Color.decode(code));  loginButton.setFocusable(false);  loginButton.addActionListener(this);  add(title);  add(usernameLabel);  add(usernameField);  add(passwordLabel);  add(passwordField);  add(loginButton);  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420-300.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 300);  add(l2);  setVisible(true);  }  public static void main(String[] args) {  new librarianlogin();  }  @Override  public void actionPerformed(ActionEvent e) {  String name = usernameField.getText();  String password = new String(passwordField.getPassword());  try {  if (e.getSource() == loginButton) {  connectSQL connect = new connectSQL();  String sql = "SELECT \* FROM librarians WHERE BINARY name = ? AND password = ?"; // Use BINARY for case-sensitive comparison  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);  statement.setString(1, name);  statement.setString(2, password);  // Execute the query  ResultSet resultSet = statement.executeQuery();  if (resultSet.next()) {  // Login successful  JOptionPane.showMessageDialog(null, "Login Successfully.", "Message", JOptionPane.INFORMATION\_MESSAGE);  setVisible(false);  new librarianSection();  } else {  // Login failed  JOptionPane.showMessageDialog(null, "Please make sure you entered correct data.", "Alert", JOptionPane.ERROR\_MESSAGE);  }  }  } catch (SQLException ex) {  ex.printStackTrace();  // Display a message indicating failure  JOptionPane.showMessageDialog(null, "An error occurred while logging in.", "Error", JOptionPane.ERROR\_MESSAGE);  }  }  } Librarian Section: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.\*;  public class librarianSection extends JFrame implements ActionListener{  private JLabel label;  private JButton addBooks, viewBooks, issueBooks, viewIssueBooks, returnBooks, logout;  public librarianSection() {  setTitle("Librarian Section");  setSize(420, 450);  setLayout(null);  setResizable(false);  setLocation(430, 150);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  label = new JLabel("Librarian Section");  label.setBounds(143, 15, 250, 50);  label.setFont(new Font("Garamond", Font.BOLD, 17));  label.setForeground(Color.black);  // Setting buttons  String code = "#F5FFFA";    addBooks = new JButton("Add Books");  addBooks.setBounds(152, 85, 110, 25);  addBooks.setFont(new Font("Arial", Font.BOLD, 10));  addBooks.setBackground(Color.decode(code));  addBooks.setFocusable(false);  addBooks.addActionListener(this);    viewBooks = new JButton("View Books");  viewBooks.setBounds(152, 130, 110, 25);  viewBooks.setFont(new Font("Arial", Font.BOLD, 10));  viewBooks.setBackground(Color.decode(code));  viewBooks.setFocusable(false);  viewBooks.addActionListener(this);      issueBooks = new JButton("Issue Book");  issueBooks.setBounds(152, 175, 110, 25);  issueBooks.setFont(new Font("Arial", Font.BOLD, 10));  issueBooks.setBackground(Color.decode(code));  issueBooks.setFocusable(false);  issueBooks.addActionListener(this);    viewIssueBooks = new JButton("View Issued Books");  viewIssueBooks.setBounds(152, 220, 110, 25);  viewIssueBooks.setFont(new Font("Arial", Font.BOLD, 8));  viewIssueBooks.setBackground(Color.decode(code));  viewIssueBooks.setFocusable(false);  viewIssueBooks.addActionListener(this);    returnBooks = new JButton("Return Book");  returnBooks.setBounds(152, 265, 110, 25);  returnBooks.setFont(new Font("Arial", Font.BOLD, 10));  returnBooks.setBackground(Color.decode(code));  returnBooks.setFocusable(false);  returnBooks.addActionListener(this);    logout = new JButton("Logout");  logout.setBounds(152, 310, 110, 25);  logout.setFont(new Font("Arial", Font.BOLD, 10));  logout.setBackground(Color.decode(code));  logout.setFocusable(false);  logout.addActionListener(this);    add(label);  add(addBooks);  add(viewBooks);  add(issueBooks);  add(viewIssueBooks);  add(returnBooks);  add(logout);    ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420x450.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 450);  add(l2);    setVisible(true);  }  public static void main(String[] args) {  new librarianSection();  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == logout) {  setVisible(false);  new AdminLogin();  } else if (e.getSource() == addBooks) {    setVisible(false);  new addBooks();    } else if (e.getSource() == viewBooks) {    setVisible(false);  new viewBooks();  } else if (e.getSource() == issueBooks) {    setVisible(false);  new issueBook();  }else if (e.getSource() == viewIssueBooks) {    setVisible(false);  new viewIssuedBooks();  }else if (e.getSource() == returnBooks) {    setVisible(false);  new returnBook();  }  }  } Add Books: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import javax.swing.\*;  public class addBooks extends JFrame implements ActionListener {  private JLabel title, callNo, name, author, publisher, quantity;  private JTextField callNofield, namefield, authorfield, publisherfield, quantityfield;  private JButton addbooks, back;  public addBooks() {  setSize(420, 450);  setResizable(false);  setLayout(null);  title = new JLabel("Add Books");  title.setBounds(150, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  callNo = new JLabel("Call No:");  callNo.setBounds(50, 100, 200, 20);  callNo.setFont(new Font("Garamond", Font.BOLD, 15));  callNo.setForeground(Color.BLACK);  name = new JLabel("Name:");  name.setBounds(50, 130, 200, 20);  name.setFont(new Font("Garamond", Font.BOLD, 15));  name.setForeground(Color.BLACK);  author = new JLabel("Author:");  author.setBounds(50, 160, 200, 20);  author.setFont(new Font("Garamond", Font.BOLD, 15));  author.setForeground(Color.BLACK);  publisher = new JLabel("Publisher:");  publisher.setBounds(50, 190, 200, 20);  publisher.setFont(new Font("Garamond", Font.BOLD, 15));  publisher.setForeground(Color.BLACK);  quantity = new JLabel("Quantity:");  quantity.setBounds(50, 220, 200, 20);  quantity.setFont(new Font("Garamond", Font.BOLD, 15));  quantity.setForeground(Color.BLACK);  callNofield = new JTextField();  callNofield.setBounds(150, 100, 200, 20);  namefield = new JTextField();  namefield.setBounds(150, 130, 200, 20);  authorfield = new JTextField();  authorfield.setBounds(150, 160, 200, 20);  publisherfield = new JTextField();  publisherfield.setBounds(150, 190, 200, 20);  quantityfield = new JTextField();  quantityfield.setBounds(150, 220, 200, 20);    String code = "#F5FFFA";    addbooks = new JButton("Add Books");  addbooks.setBounds(255, 320, 120, 30);  addbooks.setFont(new Font("Arial", Font.BOLD, 10));  addbooks.setBackground(Color.decode(code));  addbooks.setFocusable(false);  addbooks.addActionListener(this);  back = new JButton("Back");  back.setBounds(30, 320, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setFocusable(false);  back.addActionListener(this);  add(title);  add(callNo);  add(name);  add(author);  add(publisher);  add(quantity);  add(callNofield);  add(namefield);  add(authorfield);  add(publisherfield);  add(quantityfield);  add(addbooks);  add(back);  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420x450.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 450);  add(l2);  setVisible(true);  }  public static void main(String[] args) {  new addBooks();  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new librarianSection();  } else if (e.getSource() == addbooks) {  String callno = callNofield.getText();  String name = namefield.getText();  String author = authorfield.getText();  String publisher = publisherfield.getText();  String quantity = quantityfield.getText();  try {  connectSQL connect = new connectSQL();  Connection connection = connect.connection;  // Check if the book already exists in the books table  String checkQuery = "SELECT \* FROM books WHERE BINARY callno = ?";  PreparedStatement checkStatement = connection.prepareStatement(checkQuery);  checkStatement.setString(1, callno);  ResultSet resultSet = checkStatement.executeQuery();  if (resultSet.next()) {  // Book with the same callno already exists  JOptionPane.showMessageDialog(this, "Book with the same call number already exists!", "Error", JOptionPane.ERROR\_MESSAGE);  } else {  // Insert the book information into the database  String insertQuery = "INSERT INTO books (callno, name, author, publisher, quantity, issued, added\_date) VALUES (?, ?, ?, ?, ?, 0, CURDATE())";  PreparedStatement insertStatement = connection.prepareStatement(insertQuery);  insertStatement.setString(1, callno);  insertStatement.setString(2, name);  insertStatement.setString(3, author);  insertStatement.setString(4, publisher);  insertStatement.setString(5, quantity);  int rowsAffected = insertStatement.executeUpdate();  if (rowsAffected > 0) {  // Display a message indicating success  JOptionPane.showMessageDialog(this, "Book added successfully!");  } else {  // Display a message indicating failure  JOptionPane.showMessageDialog(this, "Failed to add book!", "Error", JOptionPane.ERROR\_MESSAGE);  }  insertStatement.close();  }  resultSet.close();  checkStatement.close();  connection.close();  } catch (SQLException ex) {  ex.printStackTrace();  // Display a message indicating failure  JOptionPane.showMessageDialog(this, "Failed to add book!", "Error", JOptionPane.ERROR\_MESSAGE);  }  // Clear the text fields  callNofield.setText("");  namefield.setText("");  authorfield.setText("");  publisherfield.setText("");  quantityfield.setText("");  }  }  } View Books: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import javax.swing.\*;  import javax.swing.table.DefaultTableModel;  public class viewBooks extends JFrame implements ActionListener {  private JTable table;  private JScrollPane scrollPane;  private JLabel title;  private JButton back;  public viewBooks() {  setSize(600, 400);  setResizable(false);  setLayout(null);  title = new JLabel("Books");  title.setBounds(270, 10, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  table = new JTable();  scrollPane = new JScrollPane(table);  scrollPane.setBounds(50, 60, 500, 250);    //setting buttons  String code = "#808080";    back = new JButton("Back");  back.setBounds(250, 320, 100, 30);  back.setFocusable(false);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setForeground(Color.white);  back.addActionListener(this);  add(title);  add(scrollPane);  add(back);  setVisible(true);  // Load the data from the database  loadData();  }  public static void main(String[] args) {  new viewBooks();  }  private void loadData() {  try {  connectSQL connect = new connectSQL();  String sql = "SELECT \* FROM books";  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery();  DefaultTableModel model = new DefaultTableModel();  model.addColumn("Call No");  model.addColumn("Name");  model.addColumn("Author");  model.addColumn("Publisher");  model.addColumn("Quantity");  model.addColumn("Issued");  model.addColumn("Added Date");  // Add the retrieved data to the table model  while (resultSet.next()) {  String callno = resultSet.getString("callno");  String name = resultSet.getString("name");  String author = resultSet.getString("author");  String publisher = resultSet.getString("publisher");  String quantity = resultSet.getString("quantity");  int issued = resultSet.getInt("issued");  String addedDate = resultSet.getString("added\_date");  model.addRow(new Object[]{callno, name, author, publisher, quantity, issued, addedDate});  }    resultSet.close();  statement.close();  connection.close();    table.setModel(model);  } catch (SQLException ex) {  ex.printStackTrace();  }  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new librarianSection();  }  }  } Issue Books: import java.awt.Color;  import java.util.Date;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.text.SimpleDateFormat;  import javax.swing.\*;  public class issueBook extends JFrame implements ActionListener {  private JLabel title, bookcallNo, studentId, studentName, studentContact;  private JTextField bookcallNofield, studentIdfield, studentNamefield, studentContactfield;  private JButton issuebook, back;  public issueBook() {  setSize(420, 450);  setResizable(false);  setLayout(null);  title = new JLabel("Issue Book");  title.setBounds(150, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  bookcallNo = new JLabel("Book Callno:");  bookcallNo.setBounds(50, 100, 200, 20);  bookcallNo.setFont(new Font("Garamond", Font.BOLD, 15));  bookcallNo.setForeground(Color.BLACK);  studentId = new JLabel("Student Id:");  studentId.setBounds(50, 130, 200, 20);  studentId.setFont(new Font("Garamond", Font.BOLD, 15));  studentId.setForeground(Color.BLACK);  studentName = new JLabel("Student Name:");  studentName.setBounds(50, 160, 200, 20);  studentName.setFont(new Font("Garamond", Font.BOLD, 15));  studentName.setForeground(Color.BLACK);  studentContact = new JLabel("Student Contact:");  studentContact.setBounds(50, 190, 200, 20);  studentContact.setFont(new Font("Garamond", Font.BOLD, 15));  studentContact.setForeground(Color.BLACK);  bookcallNofield = new JTextField();  bookcallNofield.setBounds(160, 100, 200, 20);  studentIdfield = new JTextField();  studentIdfield.setBounds(160, 130, 200, 20);  studentNamefield = new JTextField();  studentNamefield.setBounds(160, 160, 200, 20);  studentContactfield = new JTextField();  studentContactfield.setBounds(160, 190, 200, 20);    //setting buttons  String code = "#F5FFFA";    issuebook = new JButton("Issue Book");  issuebook.setBounds(255, 320, 120, 30);  issuebook.setFont(new Font("Arial", Font.BOLD, 10));  issuebook.setBackground(Color.decode(code));  issuebook.setFocusable(false);  issuebook.addActionListener(this);  back = new JButton("Back");  back.setBounds(30, 320, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setFocusable(false);  back.addActionListener(this);  add(title);  add(bookcallNo);  add(studentId);  add(studentName);  add(studentContact);  add(bookcallNofield);  add(studentIdfield);  add(studentNamefield);  add(studentContactfield);  add(issuebook);  add(back);    //setting image  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420x450.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 450);  add(l2);    setVisible(true);  }  public static void main(String[] args) {  new issueBook();  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new librarianSection();  } else if (e.getSource() == issuebook) {  String bookcallno = bookcallNofield.getText();  String studentId = studentIdfield.getText();  String studentName = studentNamefield.getText();  String studentContact = studentContactfield.getText();  try {  // Get the current date  Date date = new Date();  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  String issuedDate = dateFormat.format(date);  connectSQL connect = new connectSQL();  Connection connection = connect.connection;  // Check if the book call number exists in the books table  String checkQuery = "SELECT \* FROM books WHERE BINARY callno = ?";  PreparedStatement checkStatement = connection.prepareStatement(checkQuery);  checkStatement.setString(1, bookcallno);  ResultSet resultSet = checkStatement.executeQuery();  if (resultSet.next()) {  int quantity = Integer.parseInt(resultSet.getString("quantity"));  int issued = Integer.parseInt(resultSet.getString("issued"));  if (issued < quantity) {  // Book call number exists and available to issue  // Update the issued count in the books table  String updateQuery = "UPDATE books SET issued = issued + 1 WHERE callno = ?";  PreparedStatement updateStatement = connection.prepareStatement(updateQuery);  updateStatement.setString(1, bookcallno);  int rowsUpdated = updateStatement.executeUpdate();  if (rowsUpdated > 0) {  System.out.println("Issued count updated successfully.");  } else {  System.out.println("Failed to update issued count.");  }  updateStatement.close();  // Insert the issued book data into the issued\_books table  String insertQuery = "INSERT INTO issued\_books (callno, student\_id, student\_name, student\_contact, issue\_date) VALUES (?, ?, ?, ?, ?)";  PreparedStatement insertStatement = connection.prepareStatement(insertQuery);  insertStatement.setString(1, bookcallno);  insertStatement.setString(2, studentId);  insertStatement.setString(3, studentName);  insertStatement.setString(4, studentContact);  insertStatement.setString(5, issuedDate);  int rowsInserted = insertStatement.executeUpdate();  if (rowsInserted > 0) {  System.out.println("Issued book data inserted successfully.");  JOptionPane.showMessageDialog(null, "Book Issued Successfully.", "Success", JOptionPane.INFORMATION\_MESSAGE);  // Clear the text fields  bookcallNofield.setText("");  studentIdfield.setText("");  studentNamefield.setText("");  studentContactfield.setText("");  } else {  System.out.println("Failed to insert issued book data.");  }  insertStatement.close();  } else {  // All copies of the book have been issued  JOptionPane.showMessageDialog(null, "All copies of the book have been issued.", "Error", JOptionPane.ERROR\_MESSAGE);  }  } else {  // Book call number does not exist in the books table  JOptionPane.showMessageDialog(null, "Book Call Number does not exist.", "Error", JOptionPane.ERROR\_MESSAGE);  }  resultSet.close();  checkStatement.close();  connection.close();  } catch (SQLException ex) {  ex.printStackTrace();  }  }  }  } View Issued Books: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import javax.swing.\*;  import javax.swing.table.DefaultTableModel;  public class viewIssuedBooks extends JFrame implements ActionListener {  private JTable table;  private JScrollPane scrollPane;  private JLabel title;  private JButton back;  public viewIssuedBooks() {  setSize(600, 400);  setResizable(false);  setLayout(null);  title = new JLabel("Issued Books");  title.setBounds(235, 10, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  table = new JTable();  scrollPane = new JScrollPane(table);  scrollPane.setBounds(50, 60, 500, 250);    //setting buttons  String code = "#808080";  back = new JButton("Back");  back.setBounds(250, 320, 100, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setForeground(Color.white);  back.setFocusable(false);  back.addActionListener(this);  add(title);  add(scrollPane);  add(back);  setVisible(true);  // Load the data from the database  loadData();  }  public static void main(String[] args) {  new viewIssuedBooks();  }  private void loadData() {  try {  connectSQL connect = new connectSQL();  String sql = "SELECT \* FROM issued\_books";  Connection connection = connect.connection;  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery();  DefaultTableModel model = new DefaultTableModel();  model.addColumn("ID");  model.addColumn("Call No");  model.addColumn("Student Id");  model.addColumn("Student Name");  model.addColumn("Student Contact");  model.addColumn("Issue Date");  // Add the retrieved data to the table model  while (resultSet.next()) {  int id = resultSet.getInt("id");  String callno = resultSet.getString("callno");  String studentId = resultSet.getString("student\_id");  String studentName = resultSet.getString("student\_name");  String studentContact = resultSet.getString("student\_contact");  String issueDate = resultSet.getString("issue\_date");  model.addRow(new Object[]{id, callno, studentId, studentName, studentContact, issueDate});  }  resultSet.close();  statement.close();  connection.close();  table.setModel(model);  } catch (SQLException ex) {  ex.printStackTrace();  }  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new librarianSection();  }  }  } Return Book: import java.awt.Color;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.sql.PreparedStatement;  import java.sql.SQLException;  import javax.swing.ImageIcon;  import javax.swing.JButton;  import javax.swing.JFrame;  import javax.swing.JLabel;  import javax.swing.JOptionPane;  import javax.swing.JTextField;  public class returnBook extends JFrame implements ActionListener {  private JLabel title, bookcallNo, studentId;  private JTextField bookcallNofield, studentIdfield;  private JButton returnbook, back;  public returnBook() {  setSize(420, 450);  setResizable(false);  setLayout(null);  title = new JLabel("Return Book");  title.setBounds(150, 30, 200, 40);  title.setFont(new Font("Garamond", Font.BOLD, 20));  title.setForeground(Color.BLACK);  bookcallNo = new JLabel("Book Callno:");  bookcallNo.setBounds(50, 100, 200, 20);  bookcallNo.setFont(new Font("Garamond", Font.BOLD, 15));  bookcallNo.setForeground(Color.BLACK);  studentId = new JLabel("Student Id:");  studentId.setBounds(50, 130, 200, 20);  studentId.setFont(new Font("Garamond", Font.BOLD, 15));  studentId.setForeground(Color.BLACK);  bookcallNofield = new JTextField();  bookcallNofield.setBounds(150, 100, 200, 20);  studentIdfield = new JTextField();  studentIdfield.setBounds(150, 130, 200, 20);    //setting buttons  String code = "#F5FFFA";    returnbook = new JButton("Return Book");  returnbook.setBounds(255, 320, 120, 30);  returnbook.setFont(new Font("Arial", Font.BOLD, 10));  returnbook.setBackground(Color.decode(code));  returnbook.setFocusable(false);  returnbook.addActionListener(this);  back = new JButton("Back");  back.setBounds(30, 320, 120, 30);  back.setFont(new Font("Arial", Font.BOLD, 10));  back.setBackground(Color.decode(code));  back.setFocusable(false);  back.addActionListener(this);  add(title);  add(bookcallNo);  add(studentId);  add(bookcallNofield);  add(studentIdfield);  add(returnbook);  add(back);  ImageIcon imageIcon = new ImageIcon("C:\\Users\\ATECH\\Downloads\\420x450.jpg");  // Create a JLabel with the image  JLabel l2 = new JLabel(imageIcon);  l2.setBounds(0, 0, 420, 450);  add(l2);  setVisible(true);  }  public static void main(String[] args) {  new returnBook();  }  @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource() == back) {  setVisible(false);  new librarianSection();  } else if (e.getSource() == returnbook) {  String bookcallno = bookcallNofield.getText();  String studentId = studentIdfield.getText();  try {  connectSQL connect = new connectSQL();  String query = "DELETE FROM issued\_books WHERE BINARY callno = ? AND BINARY student\_id = ?";  PreparedStatement statement = connect.connection.prepareStatement(query);  statement.setString(1, bookcallno);  statement.setString(2, studentId);  int rowsDeleted = statement.executeUpdate();  if (rowsDeleted > 0) {  // Update the issued column in books table  String updateQuery = "UPDATE books SET issued = issued - 1 WHERE callno = ?";  PreparedStatement updateStatement = connect.connection.prepareStatement(updateQuery);  updateStatement.setString(1, bookcallno);  updateStatement.executeUpdate();  JOptionPane.showMessageDialog(null, "Book Returned Successfully.", "Success", JOptionPane.INFORMATION\_MESSAGE);  // Clear the text fields  bookcallNofield.setText("");  studentIdfield.setText("");  } else {  JOptionPane.showMessageDialog(null, "Book or Student ID not found.", "Error", JOptionPane.ERROR\_MESSAGE);  }  statement.close();  connect.connection.close();  } catch (SQLException ex) {  ex.printStackTrace();  }  }  }  } Connect Sql(class): import java.sql.\*;  public class connectSQL {  Connection connection;  Statement statement;  public connectSQL() {  try {  connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/libsystem", "root", "Saad8074\*\*");  statement = connection.createStatement();  } catch (SQLException e) {  e.printStackTrace();  }  }  } |

## Database work:

**1.For Storing the Record of Librarians:**

CREATE TABLE librarians (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

password VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

address VARCHAR(200) NOT NULL,

city VARCHAR(100) NOT NULL,

contactNo VARCHAR(20) NOT NULL

);

**2.For Storing the Record of Books:**

CREATE TABLE books (

id INT AUTO\_INCREMENT PRIMARY KEY,

callno VARCHAR(20) NOT NULL,

name VARCHAR(100) NOT NULL,

author VARCHAR(100) NOT NULL,

publisher VARCHAR(100) NOT NULL,

quantity VARCHAR(10) NOT NULL,

issued INT NOT NULL DEFAULT 0,

added\_date DATE NOT NULL,

INDEX idx\_callno (callno)

);

**3. For storing the record of student**

CREATE TABLE issued\_books (

id INT AUTO\_INCREMENT PRIMARY KEY,

callno VARCHAR(20) NOT NULL,

student\_id VARCHAR(20) NOT NULL,

student\_name VARCHAR(20) NOT NULL,

student\_contact VARCHAR(20) NOT NULL,

issue\_date DATE NOT NULL,

FOREIGN KEY (callno) REFERENCES books(callno)

);

## Interfaces





















