**Documentation:**

This code contains a Data Access Object (DAO) class named **MyDao**. The **MyDao** class has various methods that perform CRUD (Create, Read, Update, Delete) operations on different database tables such as the **student** table, **borrowed\_products** table, and **Item** table.

The **insertStudent()** method inserts a new student record into the **student** table, and the **insertBorrowedProduct()** method inserts a new borrowed product record into the **borrowed\_products** table. The **insertItem()** method inserts a new item record into the **Item** table.

The **updateItemValueIncrease()** method increases the quantity of an item in the **Item** table, while the **updateItemValueDecrease()** method decreases the quantity of an item in the **Item** table. The **updateItemById()** method updates an existing item record in the **Item** table based on the given item ID. The **updateItemBorrowedStatus()** method updates the status of a borrowed product record in the **borrowed\_products** table based on the given student name and product name. The **updateItemBorrowedReturnedStatus()** method updates the return status of a borrowed product record in the **borrowed\_products** table based on the given student name. The **deleteItemById()** method deletes an existing item record from the **Item** table based on the given item ID.

All the methods in the **MyDao** class take in appropriate object parameters, which contain the necessary data to perform the respective CRUD operations. The class also has a constructor that takes in a **Connection** object as a parameter, which is used to establish a connection with the database.

class named "DashBoardStudent" that extends "javax.swing.JFrame," the class for creating a GUI window. The constructor of the "DashBoardStudent" class initializes the GUI components and calls the "allItems()" method, which retrieves all the items from the database and displays them in a table.

The class also contains methods for displaying borrowed items that are either pending approval or have been approved. These methods take a user ID as a parameter, retrieve the relevant items from the database, and display them in separate tables.

There is also a method named "bill()" that creates a bill for the borrowed items. The method adds the student's name, phone number, and the current date and time to the bill, retrieves the borrowed items from a table, and displays them along with their quantities and other details in a formatted text area.

Class LogInPgaeStudent and LogInPgaeTecnician class contains several methods, including the **logInBtn1ActionPerformed**, **rSLabelImage2MouseClicked**, **nameTxtKeyPressed**, **rSLabelImage5MouseClicked**, **jCheckBox1ActionPerformed**, **passwordTxtKeyPressed**, **logInBtn2ActionPerformed**, and **validateLogIn** methods.

The **logInBtn1ActionPerformed** method is called when the user clicks on the "Sign Up" button, and it creates a new **SignUpPageStudent** object, which is then displayed on the screen. The **this.dispose()** method is called to close the current window.

The **rSLabelImage2MouseClicked** method is called when the user clicks on an image, and it creates a new **MainFrame** object, which is then displayed on the screen. The **this.dispose()** method is called to close the current window.

The **nameTxtKeyPressed** method is called when the user presses a key while the cursor is in the username text field. If the key pressed is the "Enter" key, the cursor is moved to the password field.

The **rSLabelImage5MouseClicked** method is called when the user clicks on an image, but it doesn't contain any code.

The **jCheckBox1ActionPerformed** method is called when the user clicks on the checkbox to show or hide the password. If the checkbox is checked, the password field is set to display the text, and the checkbox text is changed to "Hide Text". If the checkbox is unchecked, the password field is set to hide the text, and the checkbox text is changed to "Show Password".

The **passwordTxtKeyPressed** method is called when the user presses a key while the cursor is in the password text field. If the key pressed is the "Enter" key, the **validateLogIn()** method is called.

The **logInBtn2ActionPerformed** method is called when the user clicks on the "Log In" button. It calls the **validateLogIn()** method to ensure that both the username and password fields are filled out. If the validation is successful, it retrieves the user ID and password from the input fields, connects to the database using the **ConnectionProvider** class, and creates a **MyDao** object to interact with the database. It then retrieves the user with the given user ID and password, and checks if the user was found and the password matches. If the login is successful, it creates a new **DashBoardStudent** object, which is then displayed on the screen. The **this.dispose()** method is called to close the current window.

The **validateLogIn** method checks if both the username and password fields are filled out. If either field is empty, an error message is displayed, and the cursor is moved to the empty field. If both fields are filled out, the method returns **true.**

**MainFrame class in this class redirect to login page.**

StudentBorrowed class that retrieves data from a database table named borrowed\_products and populates two tables (itemsTablepennding and itemsTableApproved) in a user interface. The class has four methods: myBorrowedtemsPennding, myBorrowedtemsPenndingAndSearchByName, myBorrowedtemsApprove, and myBorrowedtemsApproveAndSearchName.

The myBorrowedtemsPennding method retrieves data from the borrowed\_products table where the value in the Status column is "pending" and orders the data by the dateOfReturn column in descending order. The retrieved data is then used to populate the itemsTablepennding table.

The myBorrowedtemsPenndingAndSearchByName method is similar to myBorrowedtemsPennding but includes an additional search feature where data is filtered by the studentName column based on the input in the searchNameTxt field.

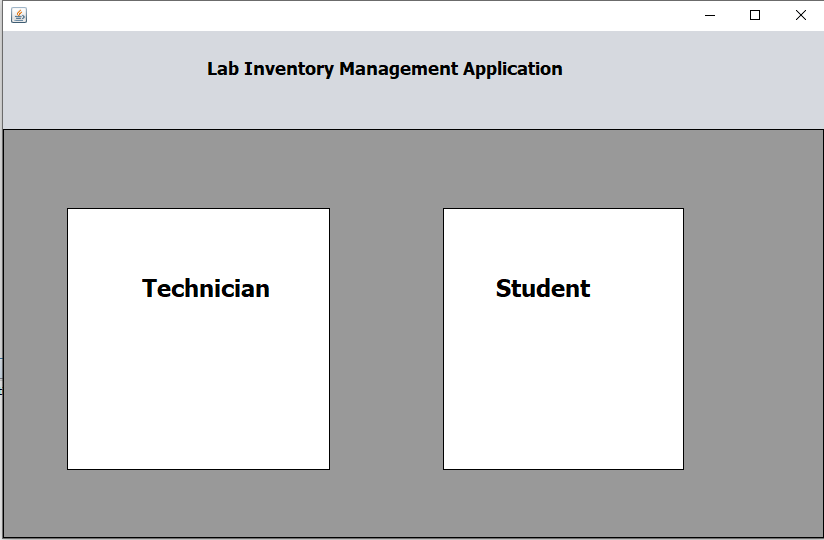
The myBorrowedtemsApprove method retrieves data from the borrowed\_products table where the value in the Status column is "handover" and orders the data by the dateOfReturn column in descending order. The retrieved data is then used to populate the itemsTableApproved table.

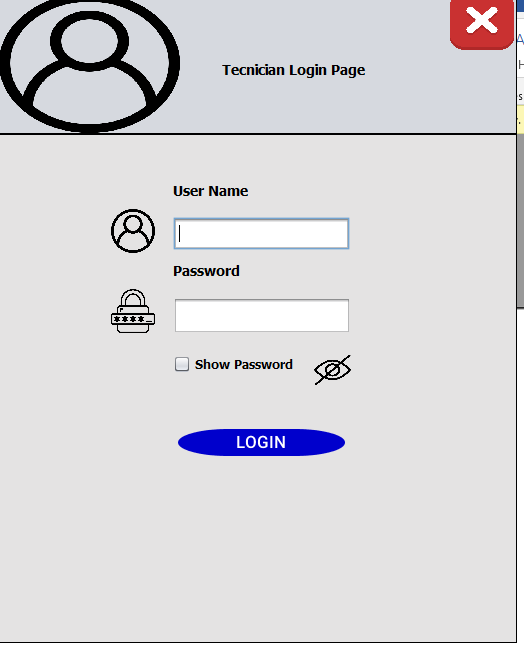
The myBorrowedtemsApproveAndSearchName method is similar to myBorrowedtemsApprove but includes an additional search feature where data is filtered by the studentName column based on the input in the searchNameTxt field.

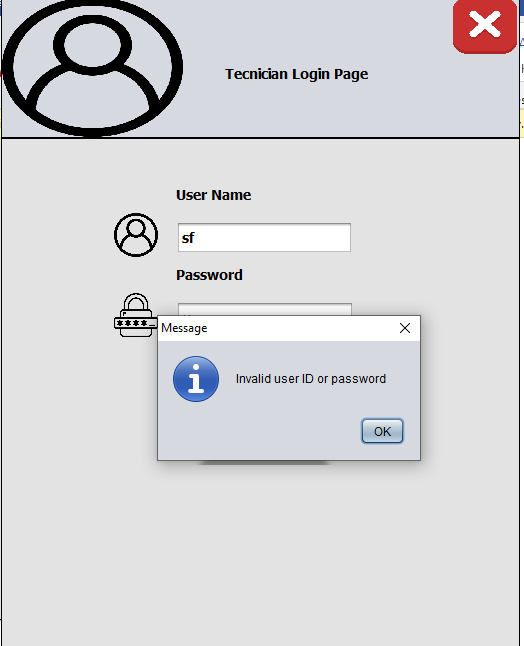
The class also has a constructor StudentBorrowed that initializes the user interface and calls the myBorrowedtemsPennding and myBorrowedtemsApprove methods to populate the two tables.

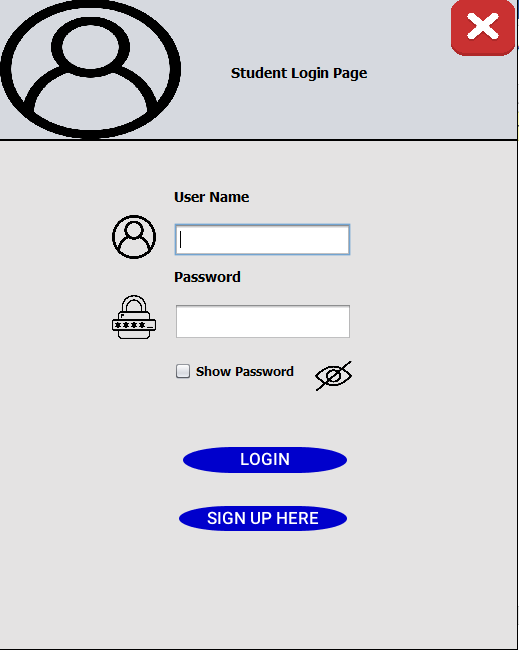
In TechnicianDashBoard class redirect to some other classes (Itom,stock , student);

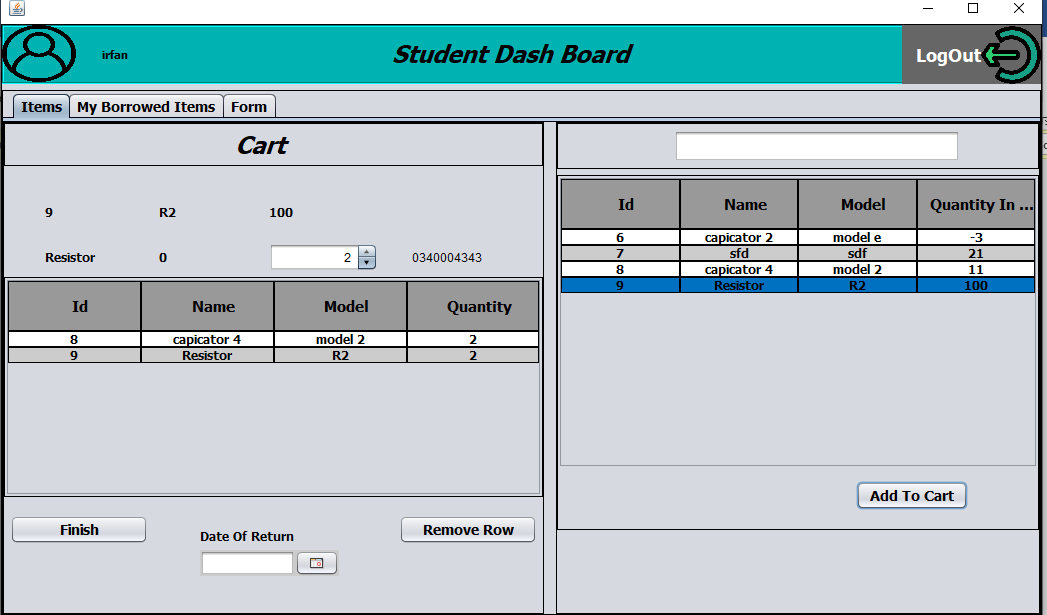
**Screenshots:**

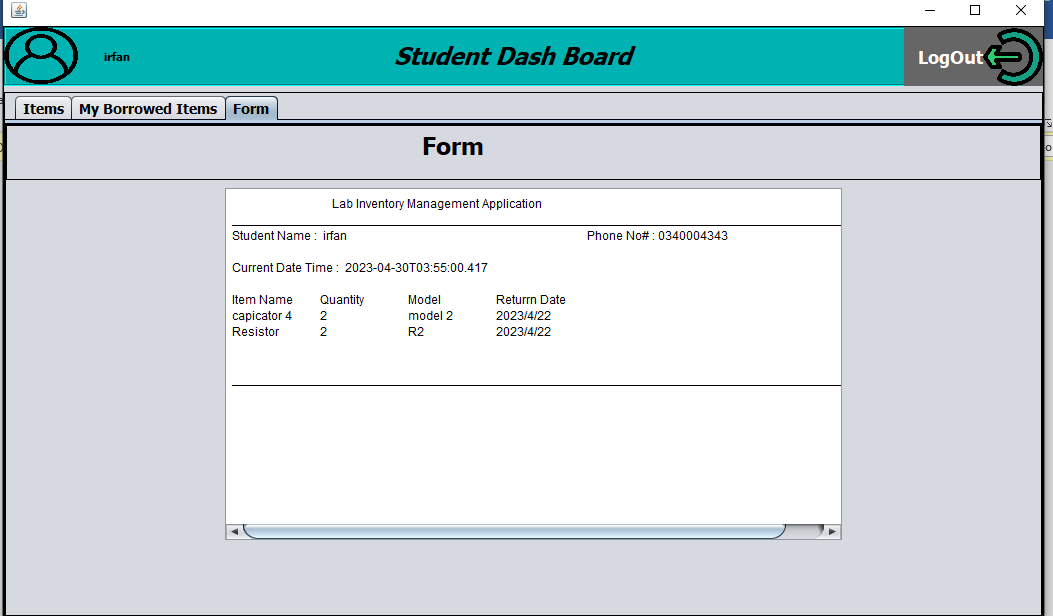


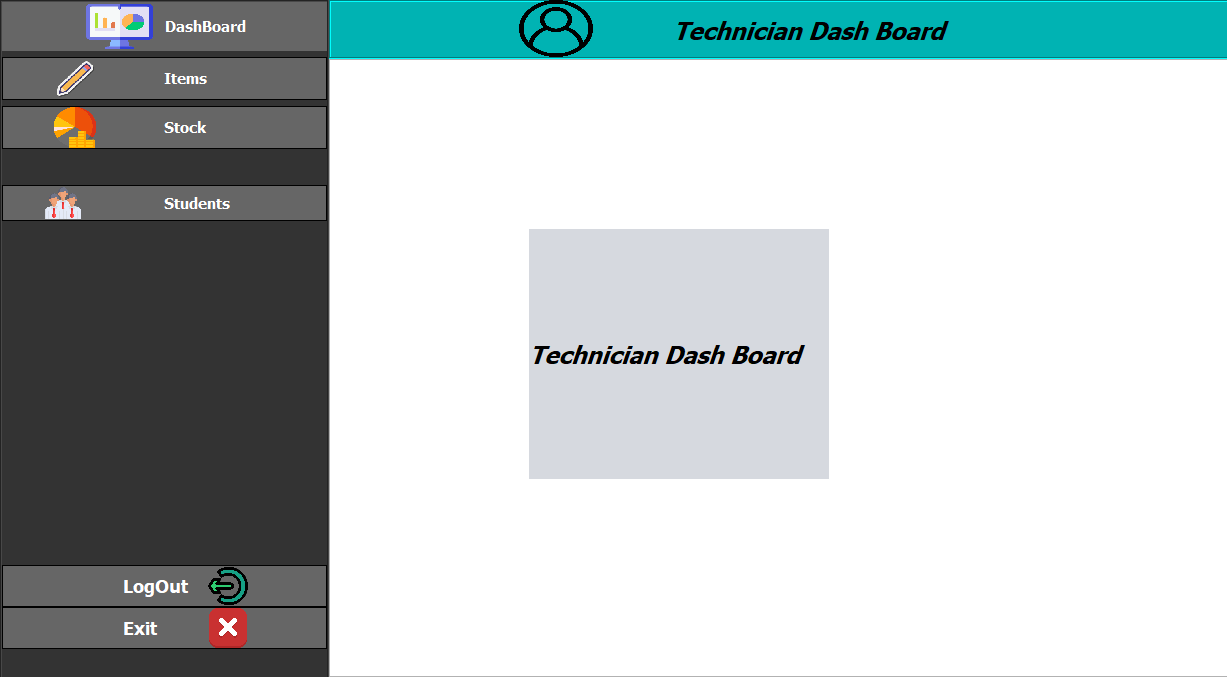


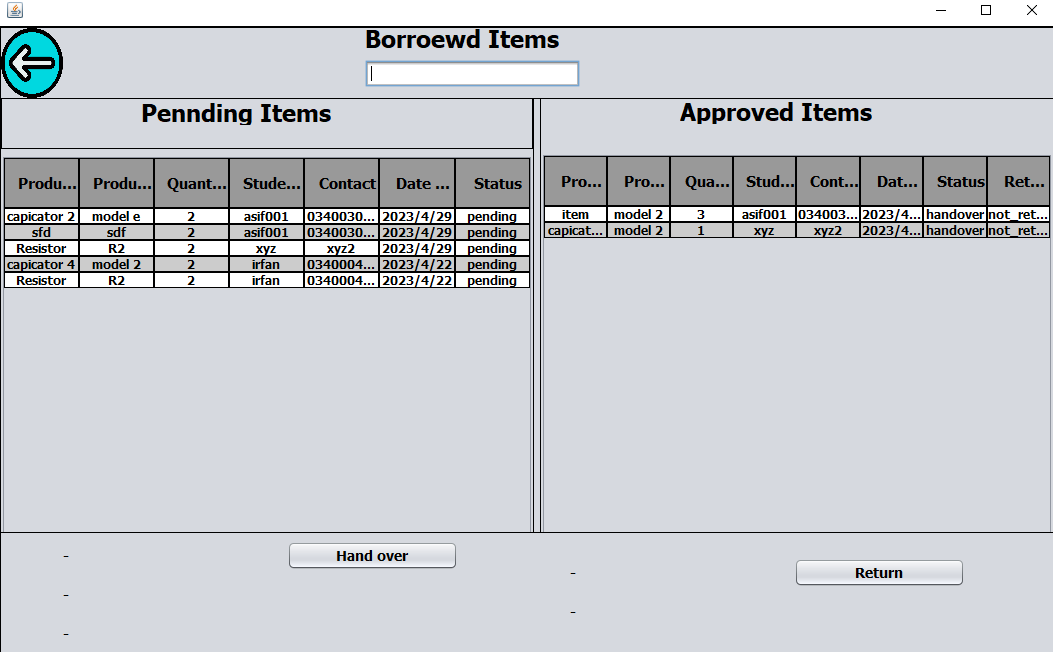












**Tools for this project:**

Xamp server and Netbeans IDE