**BOOKSHOP REGISTRATION**



SARAVANAN T

Submitted to Junnie Denny Solomon

**Summary**

* The topic of my project is a method for bookshop registration system. A list of the books will be available on the dashboard.
* Just the count needs to be added, and the book needs to be enabled.
* The registration is provided in the text area for book name, edition and the price.
* Also, the search bar is providing book details.
* Finally, there are five buttons: save, update, delete, clear and exit.

**Introduction**

In this project, we will create a bookshop registration system using Java. The program will allow the user to add, update, delete and search for books. The user will be able to input the book details such as title, edition, and price, and the program will store the data in a database.

**Tools used**

* + Java Development Kit (JDK)
  + Eclipse IDE
  + MySQL Database
  + rs2xml

**Core language**

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

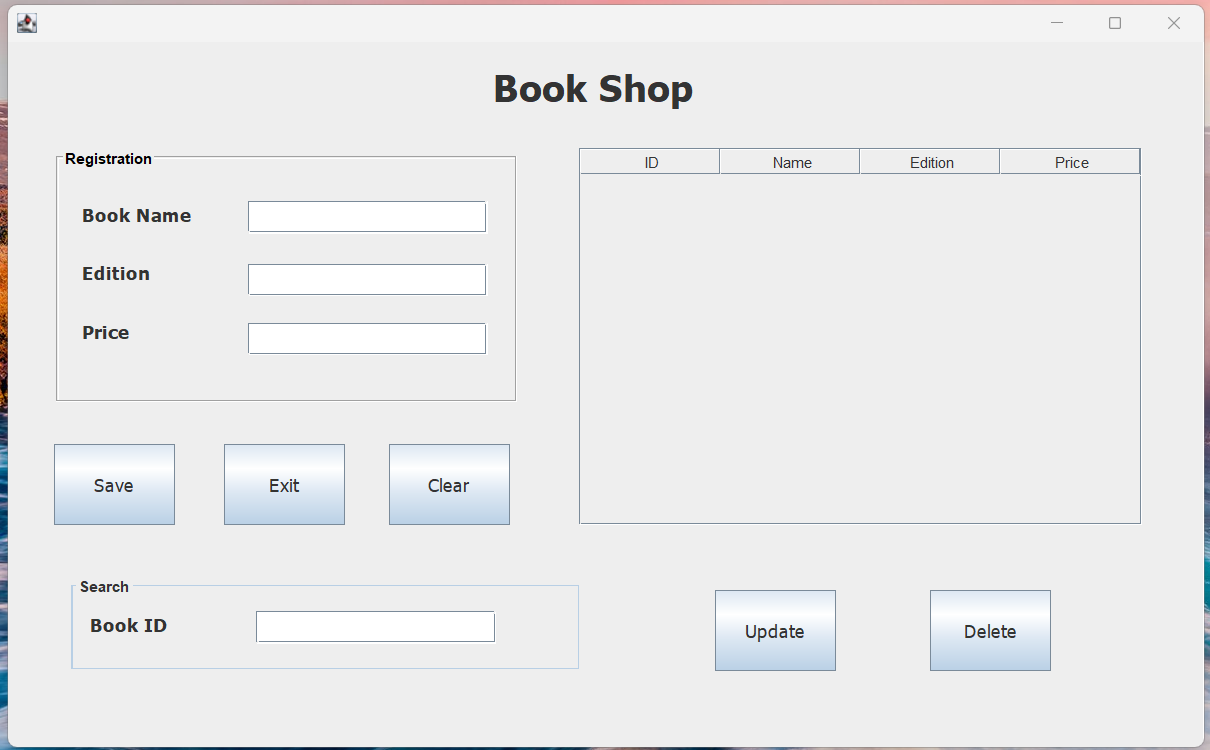
**The main Function used in java: -**

**Java swing**

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java jbutton**

The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.

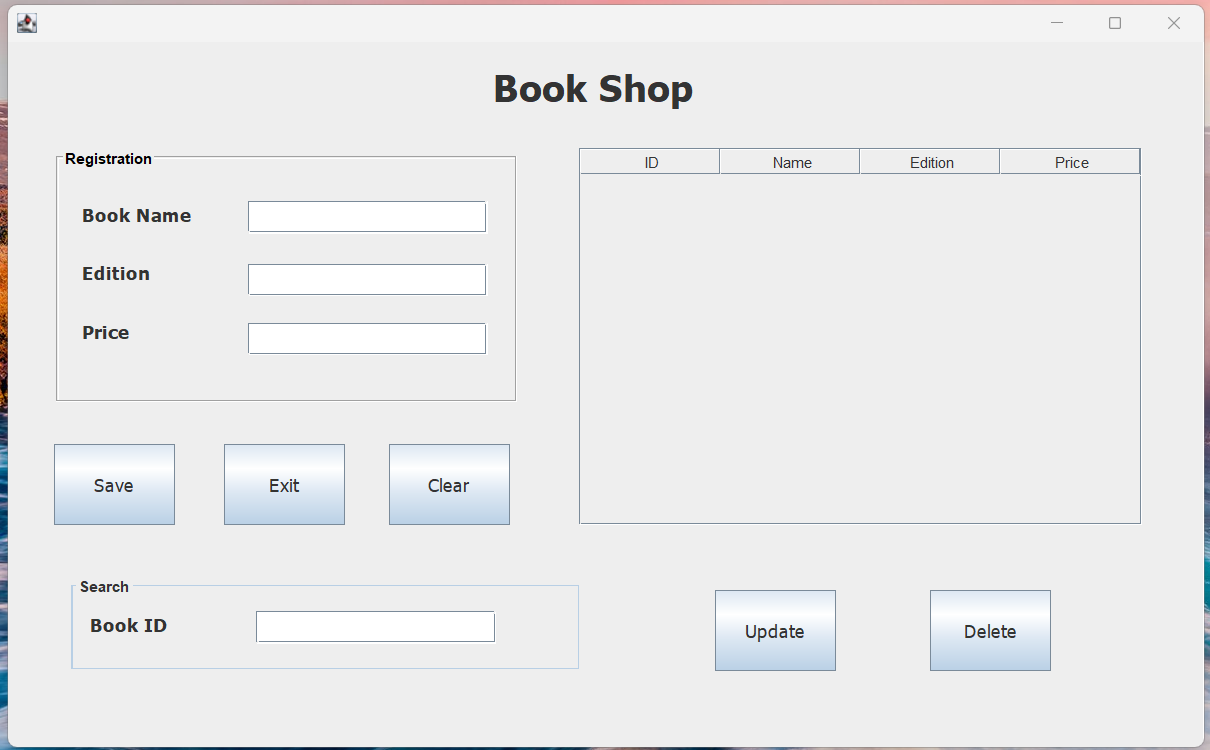


**Java jlabel**

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

**Java jtextfield**

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.



**Java jtextArea**

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class.

**Java JPanel**

The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class. It doesn't have title bar.

**Java JFrame**

The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

**Java Layout manager**

The LayoutManagers are used to arrange components in a particular manner. The **Java LayoutManagers** facilitates us to control the positioning and size of the components in GUI forms. LayoutManager is an interface that is implemented by all the classes of layout managers.

**Steps:**

* Create a Java project in Eclipse IDE.
* Add the MySQL JDBC driver to the project build path.
* Create a database in MySQL and create a table to store the book details.
* Create a Java class to represent a book and its properties.
* Create a Java class to connect to the database and perform CRUD (Create, Read, Update, Delete) operations.
* Create a Java class to handle user input and display the results.
* Test the program by running it and verifying that it works as expected.

**Implementation:**

1. Creating a Java project

* Open Eclipse IDE and create a new Java project.
* Name the project "BookShopRegistrationSystem" and click "Finish".

1. Adding the MySQL JDBC driver

* Download the MySQL JDBC driver from the official website.
* In Eclipse, right-click on the project and select "Properties".
* Click on "Java Build Path" and select the "Libraries" tab.
* Click on "Add External JARs" and select the MySQL JDBC driver.
* Click "Apply and Close".

1. Creating a database and table

* Open MySQL Workbench and connect to the database server.
* Create a new database called "bookshop" by running the following SQL query:
* CREATE DATABASE bookshop;
* Create a new table called "books" by running the following SQL query:
* CREATE TABLE books (

id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

edition INT NOT NULL,

price INT NOT NULL,

PRIMARY KEY (id)

);

1. Creating a Book class

* Create a new Java class called "Book".
* Add the following properties to the class:
* id (int)
* book name (String)
* edition (int)
* price (int)
* Generate getters and setters for all properties.

1. Creating a BookDao class

* Create a new Java class called "Book".
* Add the following methods to the class:
* public void saveBook(Book book)
* public void updateBook(Book book)
* public void deleteBook(int id)
* public Book getBookById(int id)
* public List<Book> getAllBooks()
* Implement the methods using JDBC to interact with the database.

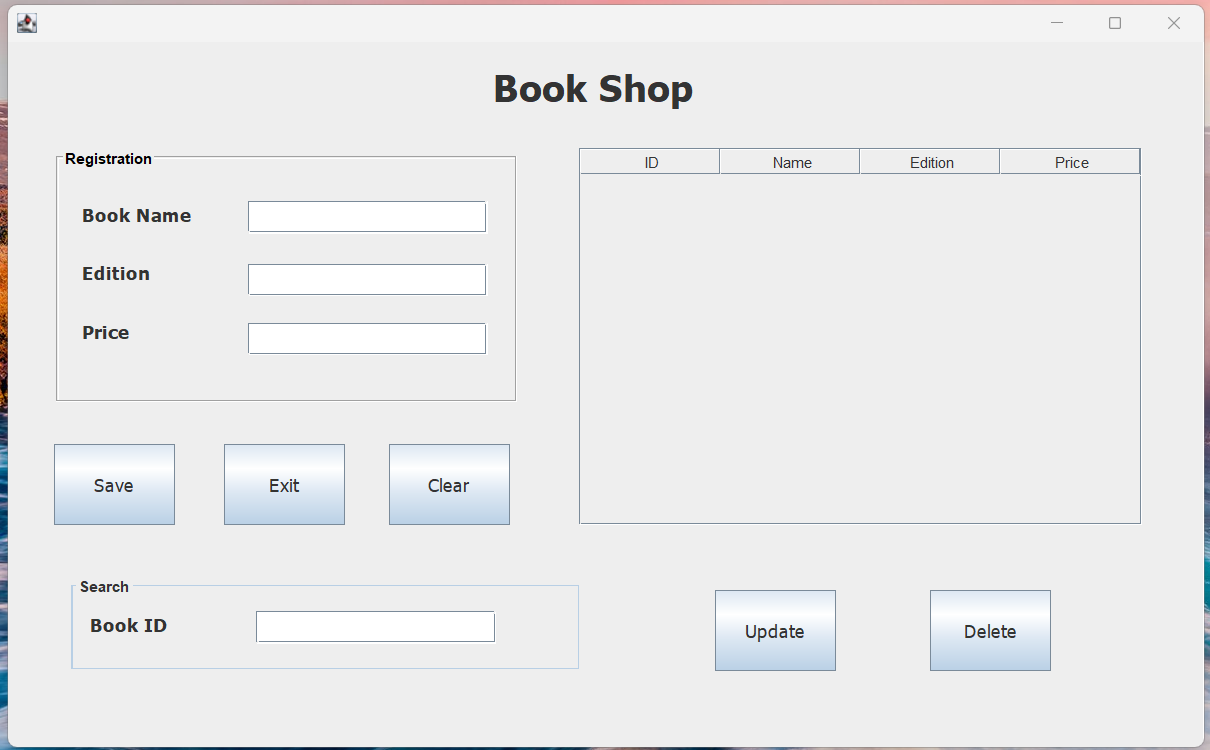
1. Creating a UserInputHandler class

* Create a new Java class called "UserInputHandler".
* Add the following methods to the class:
* public void displayMenu()
* public void handleUserInput()
* Implement the methods to display a menu to the user and handle their input.

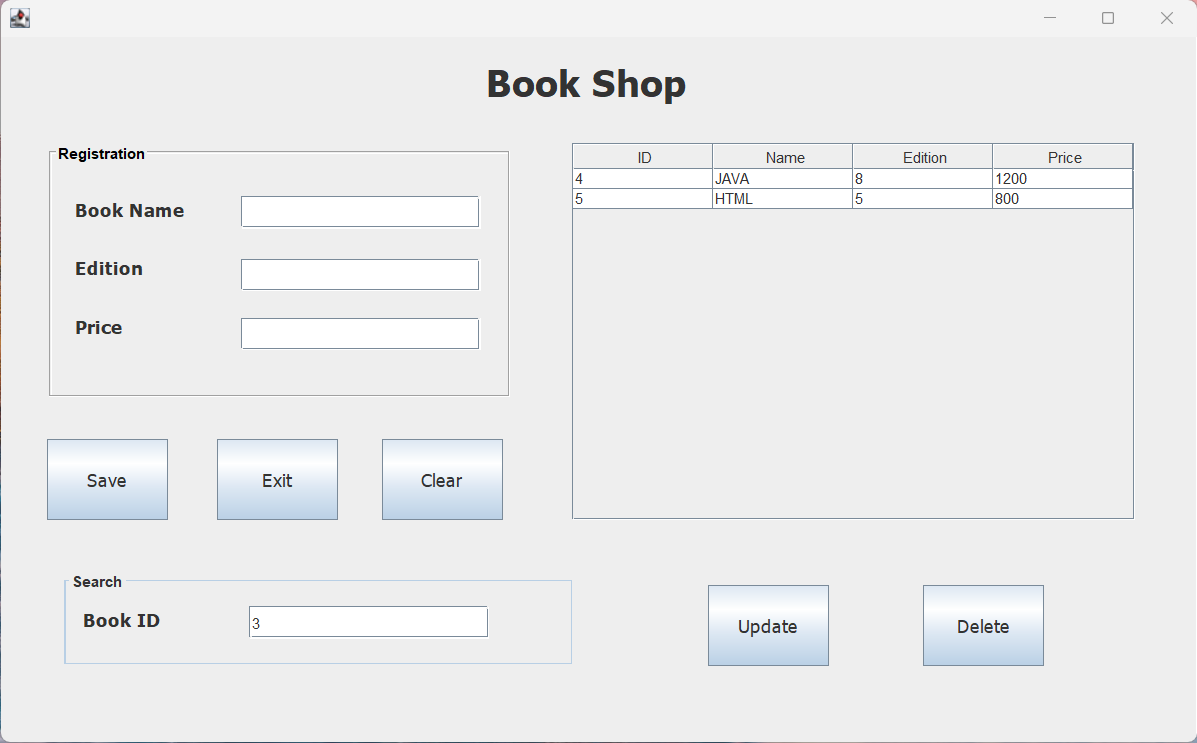
1. Testing the program

* Run the program and test each method to verify that it works as expected.

**Application preview**



**Application preview with output**



**Conclusion:**

In this project, we have created a bookshop registration system using Java. The program allows the user to add, update, delete, and search for books. The program stores the book details in a MySQL database using XAMPP.