Module Code & Module Title

CS5004NA Emerging Programming Platforms and Technologies

Assessment Weightage & Type

30% Group Coursework

Year and Semester

2020-21 Autumn

SN	Student Name	College ID	University ID
1	Surakshya Shrestha	NP01CP4S200027	19033727
2	Kushal Piya	NP01CP4S200082	19033685
3	Rohin Bharatee	NP01CP4S200077	19033708
4	Pratigya Rana	NP01CP4S200065	19033702

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

ACKNOWLEDGEMENT

Presentation, inspiration, and motivation have always played a key role in the success of any venture.

First of all, We would like to express our sincere gratitude to the advisor and lecturer, **Mr. Shreyash Mool (Module Leader), Mr. Saroj Kumar Yadav** and **Mr.Prithvi Maharjan (Module tutors)** in the **Emerging Programming Platforms and Technologies** department from whom we received comments and suggestions which helped us to carry out this project in a good way. Their patience, motivation, enthusiasm, and immense knowledge guide us a lot not only in gathering various knowledge in NetBeans project but also in all time of **discussion and analysis of development of software for an Inventory Information System.**

We are fortunate that we had the kind association as well as supervision of **London Metropolitan University** and **Islington College** for constant encouragement, painstaking efforts, and careful monitoring throughout this project for learning and grabbing opportunities in this programming world.

This group work project could not have been accomplished without the effort, support, patience, and co-operation of our group members (Surakshya Shrestha, Pratigya Rana, Kushal Piya, and Rohin Bharatee). We are grateful that we met all the requirements of this coursework and accomplished it within the submission date. Finally, yet importantly, we acknowledge with admiration the sacrifices made by our family because of our involvement with the task of completing this report. It was a great comfort and relief completing this coursework.

Table of Contents

List of Figures

1. Proposal

Title: - Bookshop Inventory Information System

1.1. Introduction

This project is all about the coursework of the module “Emerging Programming Platforms and Technologies” where the respective group is supposed to propose and develop an inventory information system according to their will. Thus, for the accomplishment of this project, we a group of four students have decided to make an information system for a **“BOOKSHOP”** which benefits both online, local, and independent bookstores where connection among readers, authors, seller, and the buyer will get strong through an inventory information system. Here, bookshop name is given as “Pustakalaya”. The task of developing an inventory information system is an excellent opportunity for us to enhance the better knowledge of how every record is being stored inside the system using GUI.

Developing an inventory information system for a bookshop is a real-life basis application that automatically helps to provide a good service to bookaholic customer regarding all types of books available in a store where users can check book history, add book records, modify & delete book records, show prices and show the availability of the particular book, etc. that certainly will bring a friendly environment to the customer while purchasing books from a retailer or wholesaler that assist local and independent bookstores financially.

This project sets aims and objectives to accomplish its respective program in a good manner where we are supposed to develop a small graphical user interface (GUI) based application that stores each detail of our respective application reliably and conveniently. With the assistance of this GUI based application, no longer the organization or store will face difficulties and errors. The main goal of this program is to create a GUI program that helps to store all the details about the books management in a store according to the customer's need for making a friendly environment, which brings a well-satisfied innovation towards a real-life foundation by saving time between buyer and seller.

Thus, by using GUI, the bookstore becomes a good inventory information system that makes buying and selling activities most comfortable and productive by bringing positive and innovative changes towards the development according to the customer needs.

1.2. Detail Information of the system

➢ List of data

Here, the bookshop information system is a GUI based application where users can input a large amount of data and information about the customer and also can store every detail about the books that have been sold and for further purpose too. The lists of data that have been used here inside our system according to the user's requirement for their convenience are mentioned below in points: -

1. **Book ID**: - Unique ID that identifies a specific book. It is stored as a string.
2. **Book Name**: Name of the book. It is stored as a string.
3. **Genre**: - The books are categorized into different genres and can be selected through combobox.
4. **Author Name**: - The name of the author. It is stored as a string.
5. **Language**: The language in which the book is written. It can be selected through radio button.
6. **Price**: Price of the book. It is stored as float.

➢ List of Features

This inventory information system is a very useful application for users in a bookstore because of its convenient, reliable, and trustworthy features and services, which bring a friendly environment of mutual understanding and sharing among users in a bookstore. Some of the features that can be seen while using our system are described below in lists: -

1. Here, inside our system, we can try **"add book records"** features that enable the user to input any type of records of the book including book name, author name, and respective data so that information about the book can be accessed easily.
2. We can also find **"show book records"** features where users can easily search history for any books that they want to see.
3. Here, **"show prices"** features can be found where users can know any prices of the book and their details too that they want to buy.
4. Here, a feature like **"check availability"** can be used by the user, which helps to find which book is sold and which one, is remaining in the bookshop.
5. Feature such as **"modify and delete book records"** where users can edit book name, author name, purchased date, etc. and they also can delete unnecessary records of the book. Similarly, deleted and modified records are also stored, which users can see whenever they want.

➢ Justification of the tools that used for development

Some of the tools we will be using in our project for the development that will help us to fulfil the requirements to accomplish our project in a good way are explained below in a short brief: -

I. NetBeans

Figure 1: NetBeans

NetBeans is an integrated development environment (IDE) for Java. It is free and open-source where users can easily develop any project using several languages such as JavaScript, C/C++, PHP, etc.

Here, for this project, we will be using this tool to carry out java programs to develop wireframes for our software using GUI. By using NetBeans, we can easily make the wireframes by dragging and dropping option so, dealing with GUI becomes easy by using this tool in comparison with Blue J. It makes making GUI easy by supporting a simplified swing application framework. Here, we can find a debugging tool that is very useful to detect the errors line by line in coding which enhances automatic correction while doing the coding. (Oracle, 2021)

Thus, NetBeans is a very useful open-source software where users can create as much as a project that they desire to create for the development of the software. Thus, this tool is recommended by our module teachers to develop GUI for our bookshop inventory information system.

II. MS Word

Figure 2: MS Word

Microsoft Word is a world-widely used word processor published by Microsoft. It was developed by Charles Simonyi and Richard Brodie and released in 1983. Microsoft Windows, Apple macOS, Android, and Apple iOS can easily access this word processor. Linux operating system can be run by MS Word using WINE. It is a software package that enables you to create, edit, print, and save documents for future retrieval and reference. It is used to create high-quality professional documents for retrieval and reference. It is used to create high-quality professionals' documents, resume, certificates, letters, reports, essay, application, etc. (Lifewire, 2021)

Thus, here we together will be using MS Word to carry out 50% of report work, which is a part of our project as we found this tool very comfortable to write any documentation work in our previous project.

III. Java

Figure 3: Java

As we know, java is an object-oriented programming language, which has been using in every corner of the world because of its fast, reliable, trustworthy, and secure features which symbolize today's IT world belongs to the java world. It is a platform-independent language so the coding of java can be run smoothly in various platforms such as Windows, Mac OS, and UNIX, etc. Thus, using java coding for the development of any type of software helps to accomplish the goal conveniently.

Here, using java-programming language, coding is going to be done in NetBeans for the development of GUI for our project.

2. Individual Tasks

For this coursework, we a group of four students (Surakshya Shrestha, Kushal Piya, Pratigya Rana, Rohin Bharatee) conducted several meetings in Google meet and we discussed together on entire process that must be carried out into our system. Here, we decided to develop an inventory system for the bookstore and bookstore name is given as “Pustakalaya”. Similarly, discussion and analysis were done cooperatively which played vital role to take many necessary steps towards the development of our inventory system. There are many tasks that should be performed in a collaboration so, we a team of four decided to divide our work so that every portion of the coursework would be finished in an appropriate way. However, we divided this coursework into parts but we used to complete our individual tasks by cooperation and discussing together in a goggle meet. Goggle meet is like a saviour for either us, which made us so easier to attend a long-hour discussion meeting among us or teachers that obviously encourage doing our activities with any obstacles during these covid days.
Here, for this coursework, most of the tasks such as discussing, analysis, planning of the system, designing of the system, making GUI, are done together cooperatively. While there are some of the individual tasks that, we completed separately are given below with a brief explanation of our each group members:-

Kushal Piya → He is the leader of our group coursework. So, he took the leadership for this coursework. Conducting meeting, arranging time, dividing individual task into partwise has done by him. Talking about individual task, the main GUI creating for the system has done by him. So, he was responsible for creating, designing and arranging all the menubar, textfield, combobox, checkbox, textarea, buttongroup for the radiobuttons etc while making GUI for the system. He has also worked on binary search algorithm. He has worked on creating table as well as inserting data into the table. Searching and sorting of the coding has also done by him.

Rohin Bharatee → He is the one who has done many research on binary search algorithm as this binary search was quite hard task, especially the coding was so difficult to implement for our system so, he as well as kushal together approached towards the solution for the binary search algorithm for our system. Rohin also completed his task of getting rid from several errors and bugs while doing the coding for our system. Validation, action performed of the coding has completed by him. He did searching on book name, author, genre etc., which is required for adding the data into the book list.

Surakshya Shrestha → She is responsible for the writing main proposal for the system describing all the tools and technologies used in the system as well as the report of the system by analysing all the parts and contents that should be present in the report. Arranging all the things in report section has done by her. She has also helped kushal and Rohin for compiling code in an appropriate way. She also worked on designing outer look of the GUI by adding background images, matching colours, logo, title etc. for making it simple but attractive. Adding functions into the table and all the testing part has done by her.

Pratigya Rana → She is the one who have helped for the method description of each method in the system. She has also worked on writing brief description about the binary search algorithm in the report section. She helped surakshya in describing the writing portion of the report as well as summarizing the report in a better way. She also worked on adding necessary comments on coding and also done documentation of the project in a better way. Gathering all the important details of the coursework and illustrating it in a brief among us has done by her.

Member Name	Activities
Kushal Piya	**- Creating the GUI. - Adding all the necessary components in the GUI. - Binary search - Inserting data - Searching and searching**
Rohin Bharatee	**- Binary search - Validation - Research on book details - Solving errors and bugs in coding - Action performed**
Surakshya Shrestha	**- Submit a proposal for the system - Designing outer look of the GUI - Compiling of the code - Adding function and testing - Documentation of the program**
Pratigya Rana	**- Method description - Documentation of the program - Brief description of the binary search in report - Conclusion**

3. Introduction

This project is all about the coursework of the second semester of the subject “Emerging Programming Platforms and Technologies”. This assignment carries 30% of the overall module weighting for our 2nd semester. This coursework must be carried out individually and we should not copy code from any source apart from the module core text and the module materials otherwise any type of unreferenced plagiarism can lead to failure of the module and suspension. So here, for the accomplishment of this project, we are recommended to take help from our module teachers.

This project sets aims and objectives to accomplish its respective program in a good manner where we are supposed to develop a small graphical user interface (GUI) based application that stores each detail of our respective application reliably and conveniently. With the assistance of this GUI based application, no longer the organization or store will face difficulties and errors. The main goal of this program is to create a GUI program that helps to store all the details about the books management in a store according to the customer’s need for making a friendly environment, which brings a well-satisfied innovation towards a real-life foundation by saving time between buyer and seller.

Thus, by using GUI, the bookstore becomes a good inventory information system that makes buying and selling activities most comfortable and productive by bringing positive and innovative changes towards the development according to the customer needs.

With many relevant researches, studies on books, websites about java programming using Swing and assistance of our module teacher, we could understand the aim and objective of this project and cope with the doubts that raised while doing this coursework such as how to get the better knowledge for developing simple and attractive software for our inventory system “bookstore”, how to apply suitable methods for programming, how to apply all the components of NetBeans while developing GUI for the system, how to get the concept of binary search algorithm and implement searching and sorting into our system, how to do testing, how to make better and attractive design for GUI, how to do coding of action performed method in a good way for the better compilation of the program etc. Thus, this coursework is easily completed through the help of our module teacher and relevant studies on our lectures slides, tutorial class and lab class.

Here, for this coursework, we have developed the inventory Java based Menu information system for the ‘bookstore” where its name is given as “Pustakalya”. For performing coding for our system, we have used NetBeans IDE and we realized this tool is so much beneficial to use and perform any type of java programming in an efficient way because by using NetBeans, we can easily make the wireframes by dragging and dropping option so, dealing with GUI becomes easy by using this tool in comparison with Blue J. It makes making GUI easy by supporting a simplified swing application framework. Here, we can find a debugging tool that is very useful to detect the errors line by line in coding which enhances automatic correction while doing the coding.

In our system, we have added many features to make our bookstore inventory system most reliable and trustworthy to use for the customers to buy any type of books by searching it according to genre, book name, and price. We can also observed many list of books available in our system with its respective genre, author, price etc. Here, menu bar containing file with open & exit submenu and also help option is added in our bookstore where a person can easily open our system as well as close the system by simply clicking on the open and exit submenu respectively. Here, an admin can easily add can add any type of book in the book list and also can delete all the booklist as well. Whenever any required field is filled left empty and if any type of invalid values, repeating data are entered then, the system automatically shows a dialogue box with an appropriate message such as duplicate message, invalid data error etc. Here, validation, searching and sorting is done based on binary search algorithm to make the system more reliable to search the books according to the price, book name and genre. If any user does not know about how to use the system then there is help option, which shows available features and information, rules and regulations of the system.

In this way, we have included all the important parts in our system and implement in a logic way. Here, we have also attached testing portion where all the features of the system are tested and these testing are presented here with screenshots. Each brief description with proper explanation of the binary search algorithm, sorting and searching is also presented here.

4. Binary Search

Binary search is a quick and effective search algorithm with a working principle of divide and conquer which is used to detect the position of a specific value from a sorted array. Binary search first start to work by searching in the middle of the array and then going down the first lower or upper half of the sequence. If the middle value or the median is lower than the required value(target), then the search goes to the higher half, but if the median value is higher than the target then the search needs to look on the descending portion of the array (techopedia, January 18, 2017).

4.1. Working mechanism of Binary search Algorithm

For the Binary search to work, the array should be in sorted form i.e. either in ascending order or in descending order. For example, let us have an array arr = {11, 12, 13, 14, 15, 16, 17, 18, 19} which is arranged in ascending order and the target value will be 17 i.e. the location of the item to be searched is 17.

We have a formula:

Mid = (high + low) /2

Where,

Low = 0

High = Size of the array

Step 1:

11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8

Here, Low = 0 and High = 8

Now,

Mid = (high + low) /2

= (8 + 0) /2 = 4

If we compare the value stored at location 4, with the target value, they do no match as the value stored at location 4 is 15 which are lesser than the target value 17. Now the search goes towards the higher half to the right of the middle item. Now, for the step 2 the low will be mid + 1 as the mid was found to be lesser than the target value.

Step 2:

11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8

Here, Low = mid + 1 = (4 +1) = 5, High = 8

Mid = (high + low) /2 = (8 + 5)/2 = 6 (integer value of 6.5)

Therefore, we conclude the target value 17 is stored at location 6.

4.2. Implementation of Binary search in the program

On this project, Binary search algorithm was used in order to search the items based on the price. At first, a loop was run to fetch all the data from the rows of jtable and that loop was continued and total data were stored in data array. Before performing the Binary search, the data array was sorted. After that, the item which was to be searched was compared with the middle value of the list, if they matched then it returns the index of the middle value. If they did not match, then it further checks if the searched item is smaller or greater than the middle value. If the searched item was found to be smaller than the middle value, the action was repeated on the subarray of the left-side of the middle value. And if the searched item was found to be greater than the middle value, the action was repeated on the subarray of the right-side of the middle value. And if the remaining array to be searched was found to be empty then a “not found’’ message used to display.

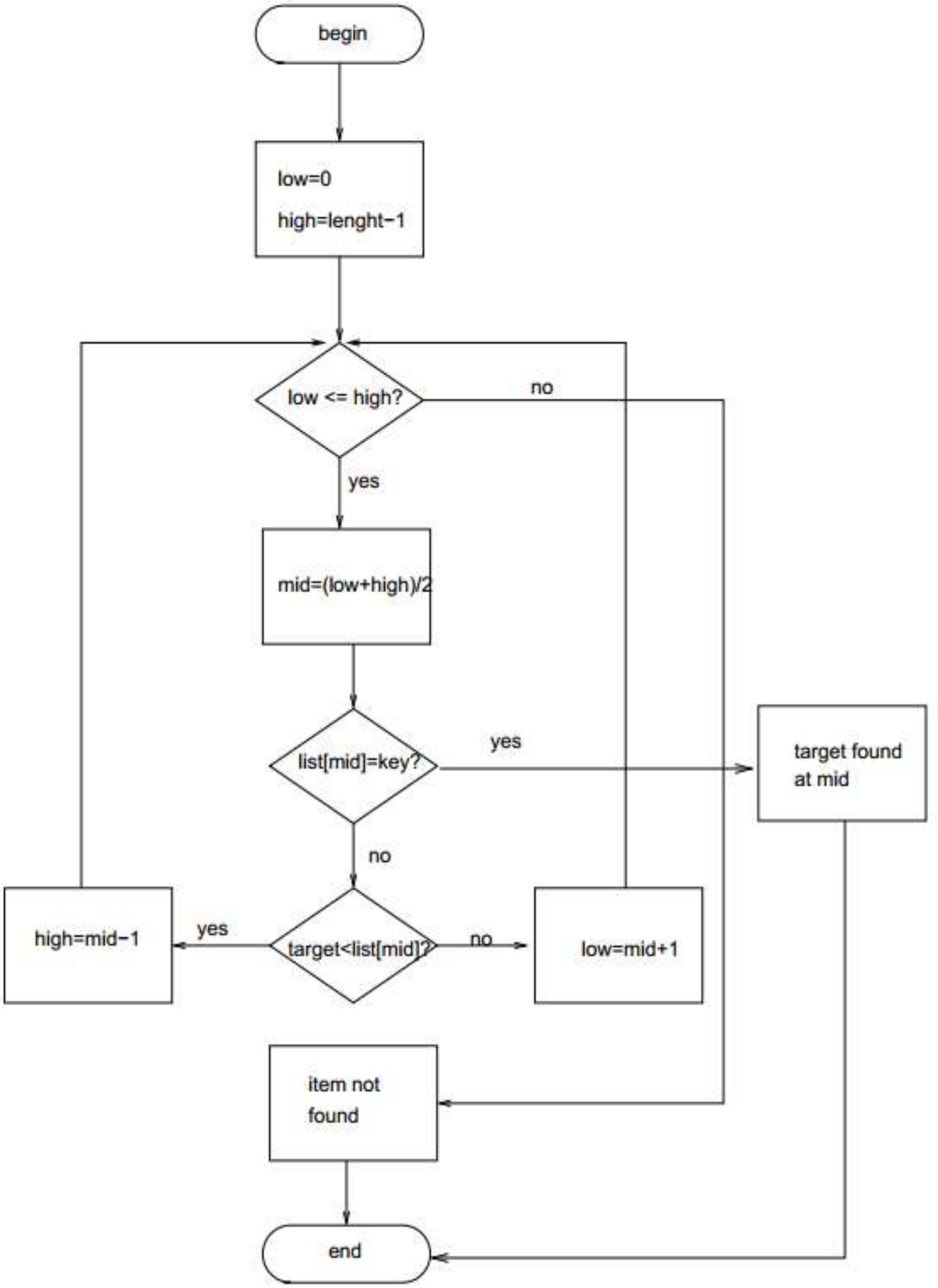


Figure 4: Binary search algorithm flowchart

5.Sorting Algorithm Name

A Sorting Algorithm is used to rearrange the elements of a given array or list according to an element comparison operator. In the respective data structure, the comparison operator is used to decide the new element order. (GeeksforGeeks, 2020)
For example: The list of characters below is sorted by their ASCII values in an increasing order. In other words, characters with a lower ASCII value are put first than those with a higher ASCII value.

I S L I N G T O N C O L L E G E ➔ C E E G G I I L L L N N O O S T

Input Output

5.1 Working Mechanism of the (selected sorting technique)

For our program, we have used Binary Insertion sort. Insertion sort is an in-place online stable sorting algorithm that generates one item at a time for the final sorted list. This works on the concept of moving an item in a sorted array to its correct location. Similarly, we can use binary search to reduce the number of comparisons in normal insertion sort. Binary Insertion Sort uses binary search to find the proper location to insert the selected item at each iteration.

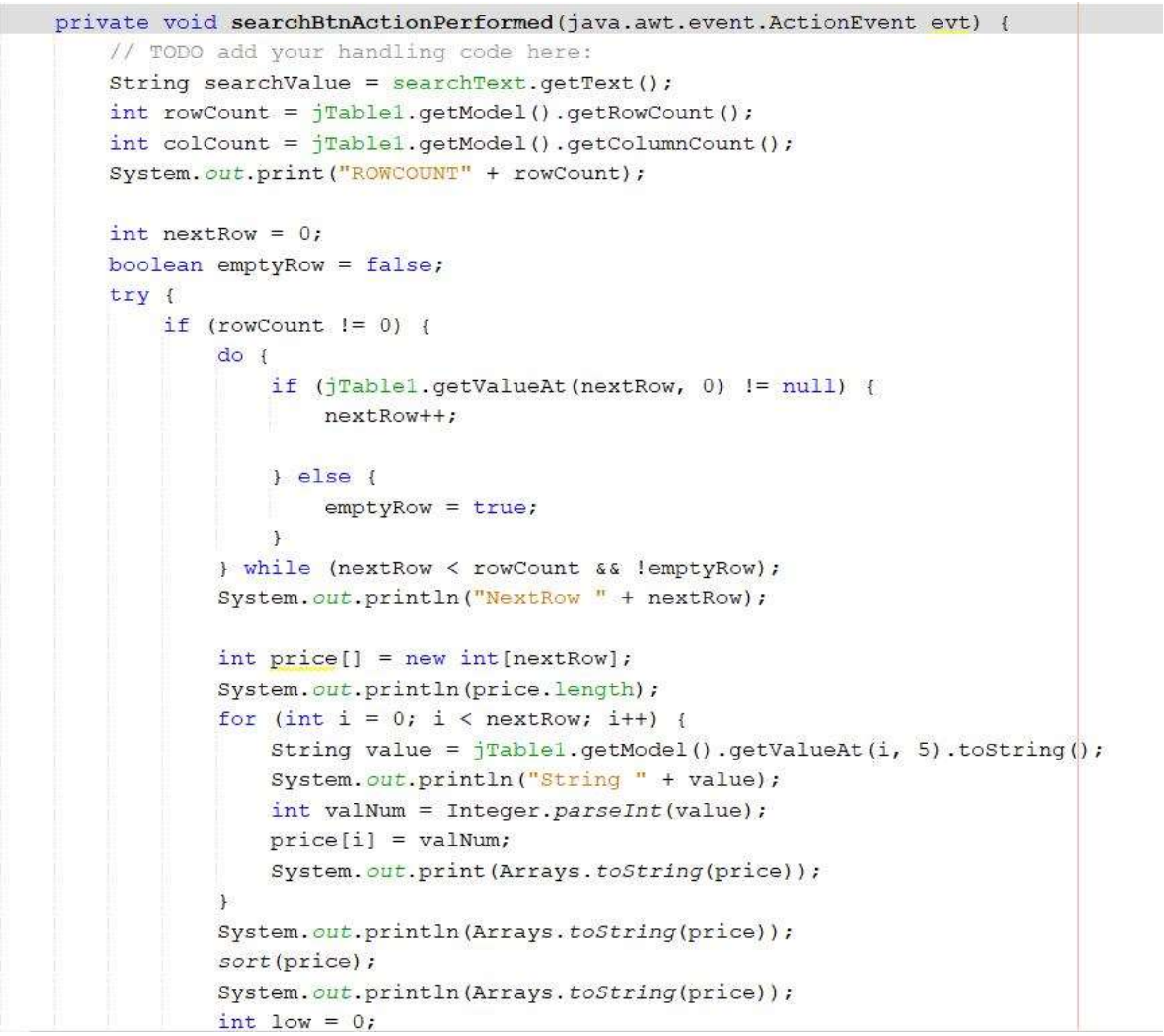
In normal insertion sort, it takes O(n) comparisons (at nth iteration) in the worst case. We can reduce it to O(log n) by using binary search. (GeeksforGeeks, 2020)

In short, Binary search is used in Insert Sort to minimize the number of comparisons. This shift is known as Sort Binary Insertion.

Binary Insertion Sort uses binary search to find the correct position for each iteration to insert the selected object. In the insertion form, in the worst case, O(i) (ith iteration) is taken. It is reduced to O using binary search (log i). (Ryder, 2019)

5.2 Implementation of (selected sorting technique) in program

In our project, Binary Insertion Sort was implemented in the following manner:-



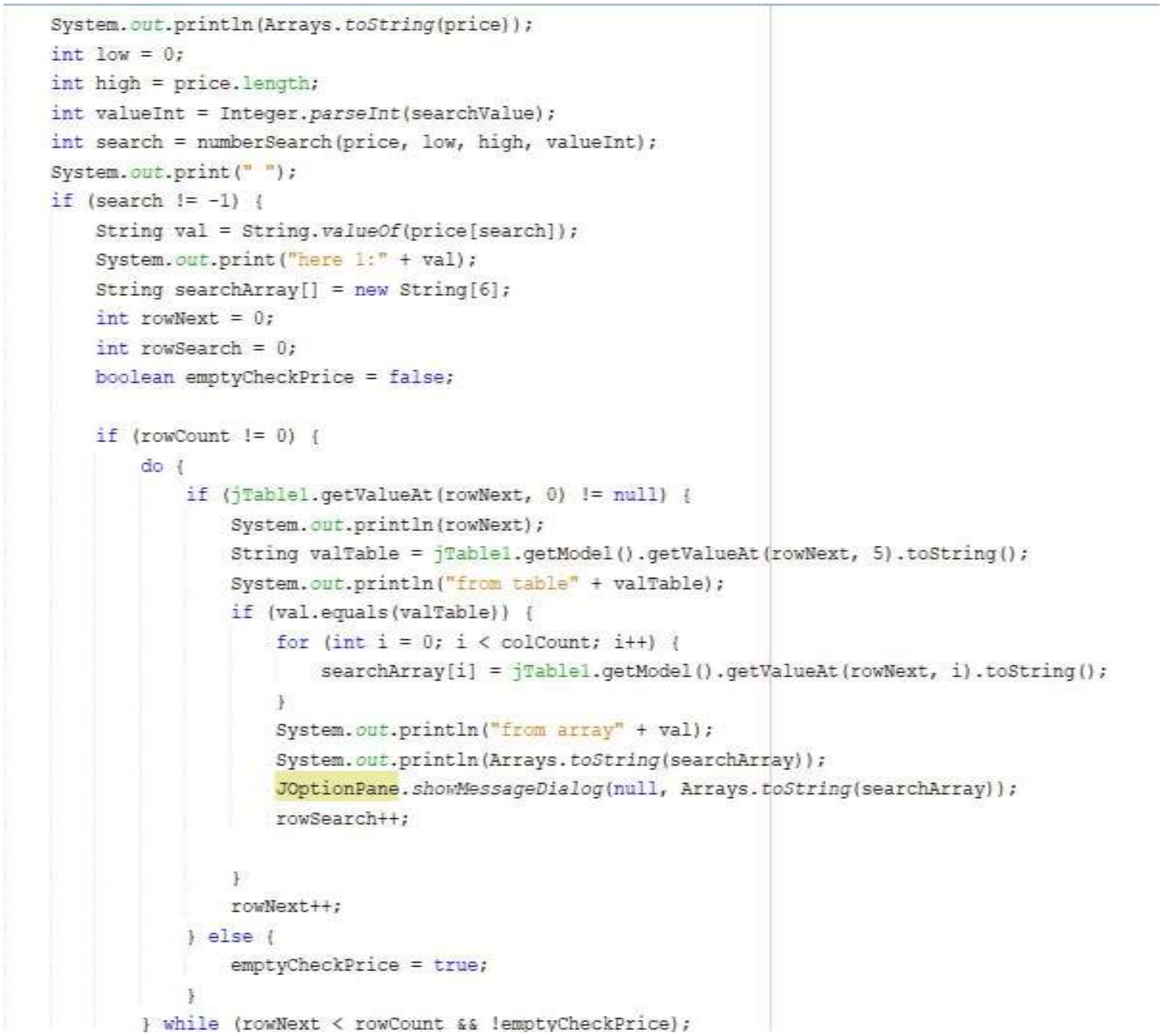
```
private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String searchValue = searchText.getText();
    int rowCount = jTable1.getModel().getRowCount();
    int colCount = jTable1.getModel().getColumnCount();
    System.out.print("ROWCOUNT" + rowCount);

    int nextRow = 0;
    boolean emptyRow = false;
    try {
        if (rowCount != 0) {
            do {
                if (jTable1.getValueAt(nextRow, 0) != null) {
                    nextRow++;

                } else {
                    emptyRow = true;
                }
            } while (nextRow < rowCount && !emptyRow);
            System.out.println("NextRow " + nextRow);

            int price[] = new int[nextRow];
            System.out.println(price.length);
            for (int i = 0; i < nextRow; i++) {
                String value = jTable1.getModel().getValueAt(i, 5).toString();
                System.out.println("String " + value);
                int valNum = Integer.parseInt(value);
                price[i] = valNum;
                System.out.print(Arrays.toString(price));
            }
            System.out.println(Arrays.toString(price));
            sort(price);
            System.out.println(Arrays.toString(price));
            int low = 0;
```

Figure 5 :Implementation of sorting1.1

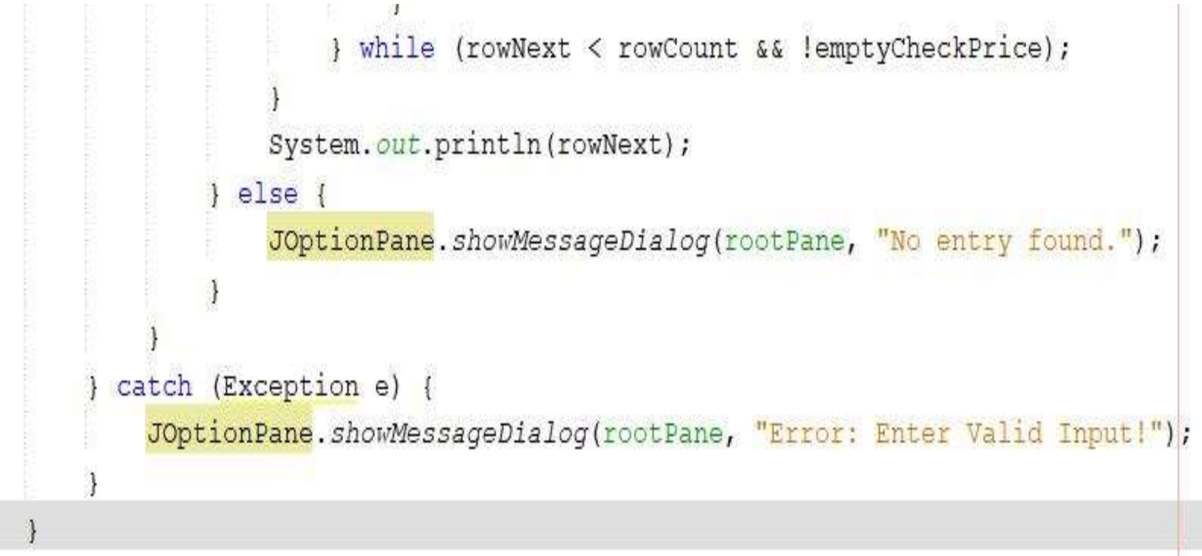


```
System.out.println(Arrays.toString(price));
int low = 0;
int high = price.length;
int valueInt = Integer.parseInt(searchValue);
int search = numberSearch(price, low, high, valueInt);
System.out.print(" ");
if (search != -1) {
    String val = String.valueOf(price[search]);
    System.out.print("here 1:" + val);
    String searchArray[] = new String[6];
    int rowNext = 0;
    int rowSearch = 0;
    boolean emptyCheckPrice = false;

    if (rowCount != 0) {
        do {
            if (jTable1.getValueAt(rowNext, 0) != null) {
                System.out.println(rowNext);
                String valTable = jTable1.getModel().getValueAt(rowNext, 5).toString();
                System.out.println("from table" + valTable);
                if (val.equals(valTable)) {
                    for (int i = 0; i < colCount; i++) {
                        searchArray[i] = jTable1.getModel().getValueAt(rowNext, i).toString();
                    }
                    System.out.println("from array" + val);
                    System.out.println(Arrays.toString(searchArray));
                    JOptionPane.showMessageDialog(null, Arrays.toString(searchArray));
                    rowSearch++;

                }
                rowNext++;
            } else {
                emptyCheckPrice = true;
            }
        } while (rowNext < rowCount && !emptyCheckPrice);
```

Figure 6: Implemenation of sorting 1.2



```
                    } while (rowNext < rowCount && !emptyCheckPrice);
                }
                System.out.println(rowNext);
            } else {
                JOptionPane.showMessageDialog(rootPane, "No entry found.");
            }
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(rootPane, "Error: Enter Valid Input!");
    }
}
```

Figure 7:Implementation of sorting 1.3

Hence, the above figures show how the binary insertion sorting was used as a sorting algorithm in the given project.

6. Method Description

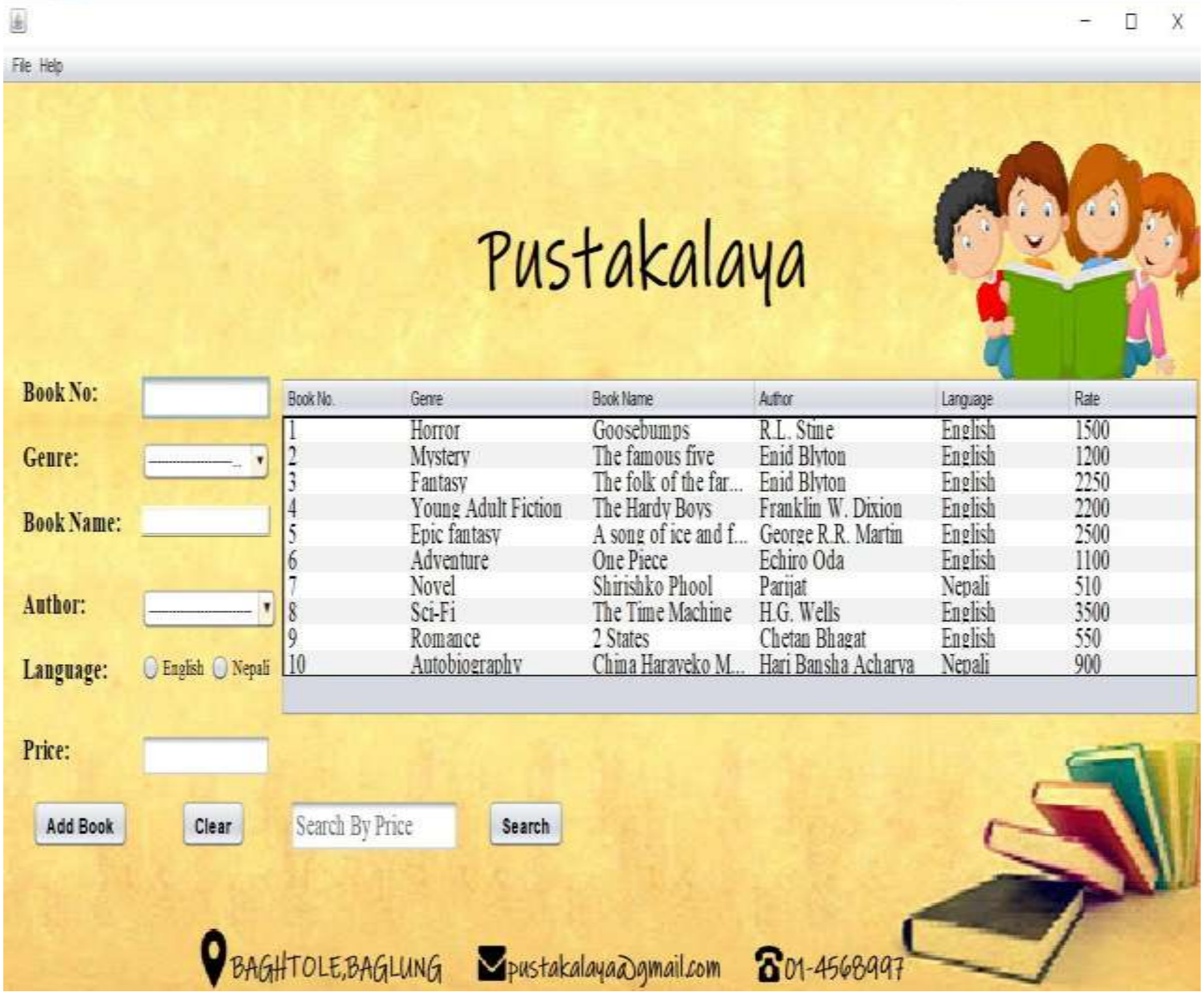
Method	Description
Bookinfo( )	It is used to initialize components. Dataadd( ) method is also called here after creating a default table model
boolean noDuplicateData(String productModel, int index)	It is used to pass String and Integer value which are used to check the duplicate data.
dataadd()	This method is used to add inserted value to the default table model
int noSearch(int[ ] a, int low, int high, int value)	In this method, the binary search action is performed.
swap(int[ ] a, int i, int j)	This method is used for swapping the values while sorting
sort(int[ ] a)	This method is used for sorting the price
int minimumPosition(int[ ] a, int from)	This method receives the value from the sorting and then calls the minimumPosition
int searchString(String[ ] arr, String x)	This method is used for searching string. In our project, the search is done from the name of the book.

searchBtnActionPerformed ( )	This method is used to display the information of the searched book.
jMenuItem1ActionPerformed (	This method is used to open existing file.
jMenuItem2ActionPerformed ( )	This method is used to terminate the program
searchTextFocusLost ( )	This method is used to keep the placeholder.
searchTextFocusGained ( )	This method is used to remove the placeholder
addBookBtnActionPerformed( )	This method is sued to add the data in the table
clearBtnActionPerformed ( )	This method is used to clear the textfield
main(String args[ ])	This method is used to execute the program

7. Testing

7.1 Test 1: Running the program in NetBeans

Objective	➢ **To check the java programs whether it can be complied and run in NetBeans for opening bookstore inventory information system main window.**
Action	➢ **At first, the bookstore project is opened.** ➢ **Then run button (fn+ shift + f6) is pressed to run the program.**
Expected Result	➢ **The whole program must be compiled and run successfully.** ➢ **The main GUI window must be executed.**
Actual Result	➢ **The whole program has been compiled and ran successfully.**
Conclusion	➢ **The test was carried out successfully.**

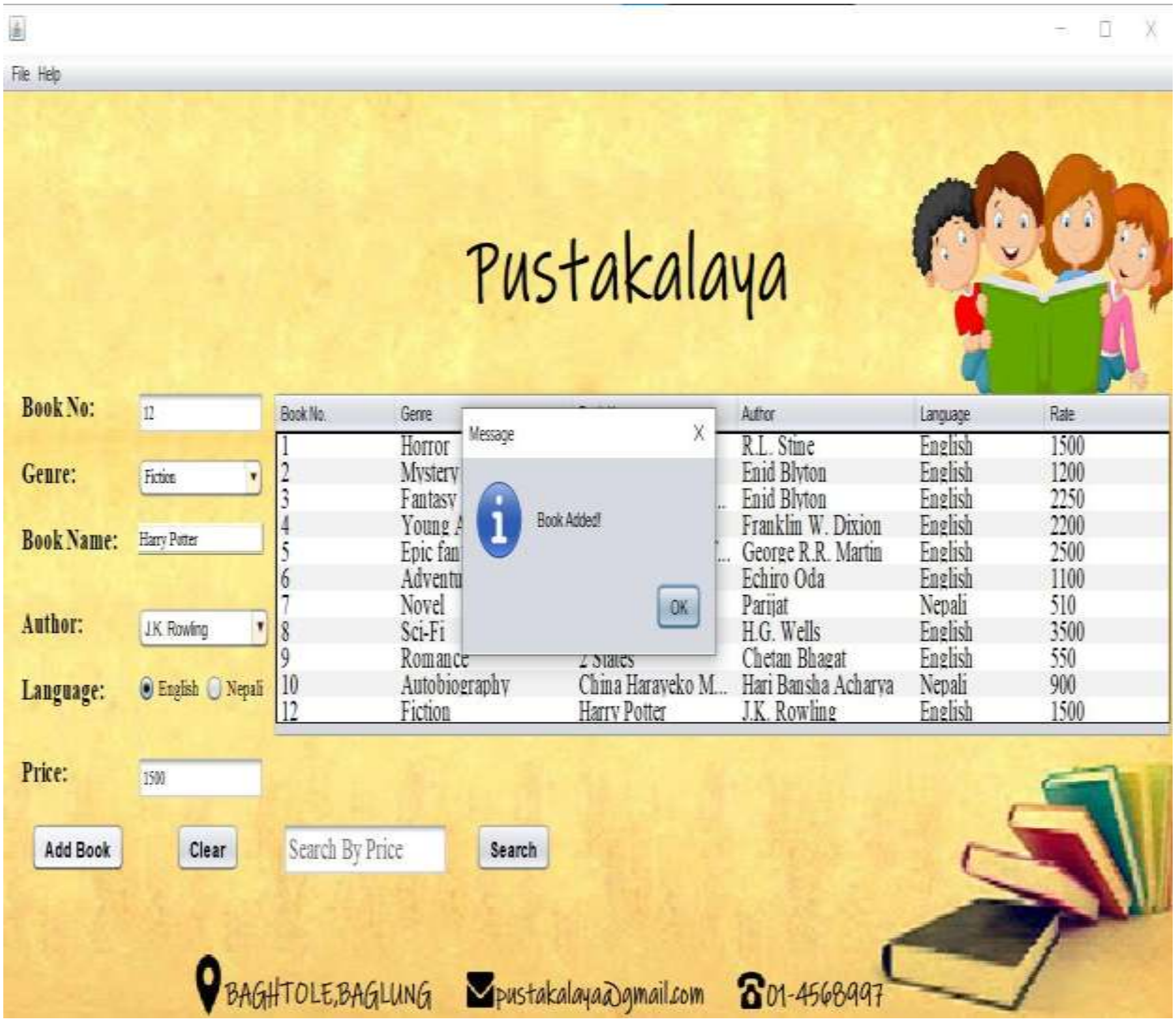


Book No.	Genre	Book Name	Author	Language	Rate
1	Horror	Goosebumps	R.L. Stine	English	1500
2	Mystery	The famous five	Enid Blyton	English	1200
3	Fantasy	The folk of the far...	Enid Blyton	English	2250
4	Young Adult Fiction	The Hardy Boys	Franklin W. Dixion	English	2200
5	Epic fantasy	A song of ice and f...	George R.R. Martin	English	2500
6	Adventure	One Piece	Echiro Oda	English	1100
7	Novel	Shirishko Phool	Parijat	Nepali	510
8	Sci-Fi	The Time Machine	H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Haraveko M...	Hari Bansha Acharya	Nepali	900

Figure 8: Result of test 1

7.2 Test 2: Adding book details in the list

Objective	➢ **To check the java programs in NetBeans whether it can be used to add the book details in GUI.**
Action	➢ **At first, the bookstore GUI is opened by clicking on run button.** ➢ **Then, the values for book no, genre, book name, author, language, price are entered.** ➢ **The "ADD BOOK" button is clicked.** ➢ **The list of book with its respective details is observed automatically after clicking on "Add book button".**
Expected Result	➢ **The added book details must be stored in the list in the side table.**
Actual Result	➢ **The whole program has been compiled and ran successfully and added book list is observed in the table.**
Conclusion	➢ **The test was carried out successfully.**



Book No: 12

Genre: Fiction

Book Name: Harry Potter

Author: J.K. Rowling

Language: ● English ○ Nepali

Price: 1500

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy		Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan		George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Harayeko M...	Hari Bansha Acharya	Nepali	900
12	Fiction	Harry Potter	J.K. Rowling	English	1500

Message

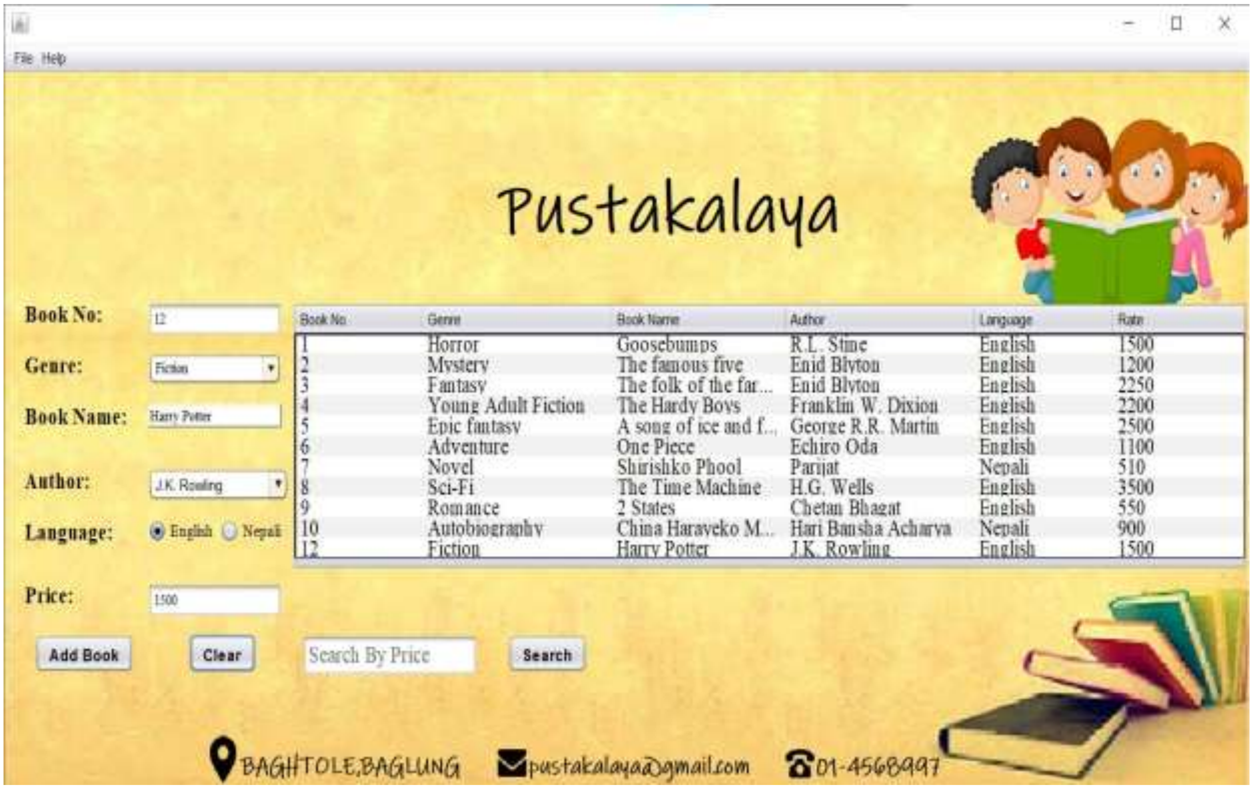
Book Added!

BAGHTOLE,BAGLUNG pustakalaya@gmail.com 01-4568997

Figure 9:Result of test 2

7.3 Test 3: Clear data

Objective	➢ **To check the java programs in NetBeans whether it can be used to clear the data.**
Action	➢ **At first, all the textfields are filled** ➢ **Then, the clear button is pressed**
Expected Result	➢ **All the textfields should be reset**
Actual Result	➢ **The textfields were reset**
Conclusion	➢ **The test was carried out successfully.**



Book No: 12

Genre: Fiction

Book Name: Harry Potter

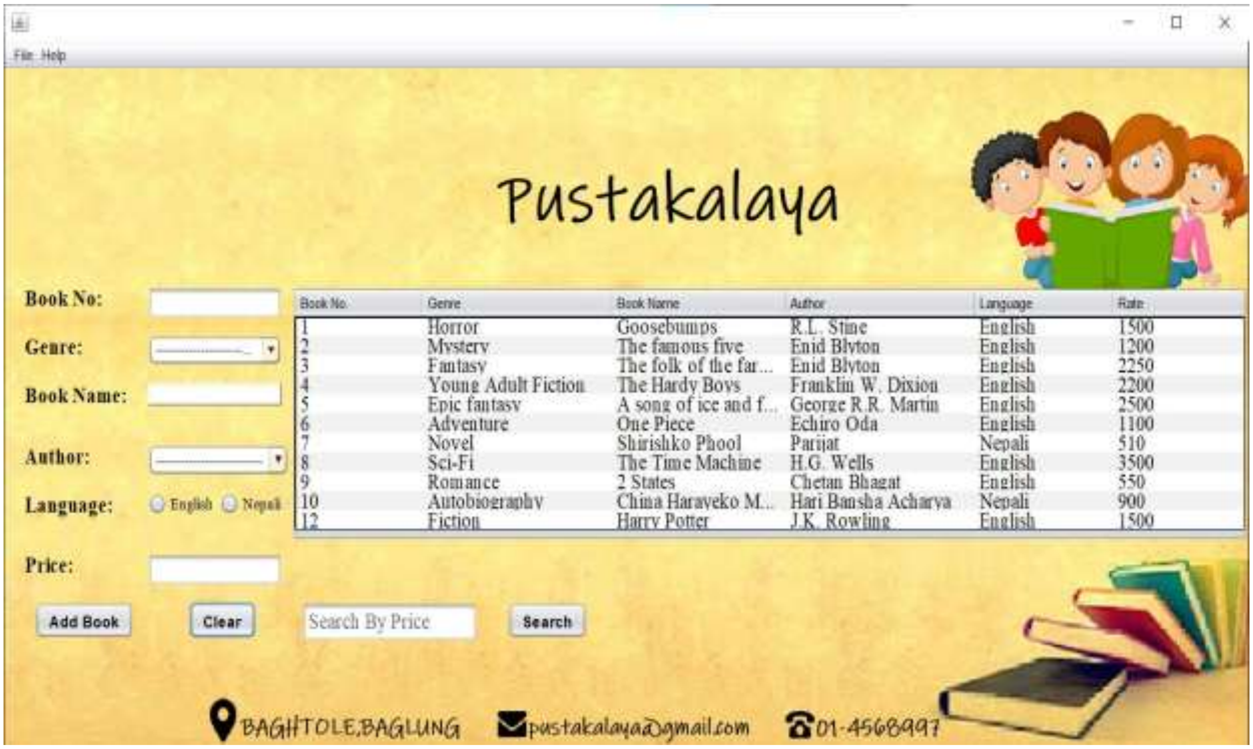
Author: J.K. Rowling

Language: ● English ○ Nepali

Price: 1500

Book No	Genre	Book Name	Author	Language	Rate
1	Horror	Goosebumps	R.L. Stine	English	1500
2	Mystery	The famous five	Enid Blyton	English	1200
3	Fantasy	The folk of the far...	Enid Blyton	English	2250
4	Young Adult Fiction	The Hardy Boys	Franklin W. Dixion	English	2200
5	Epic fantasy	A song of ice and f...	George R.R. Martin	English	2500
6	Adventure	One Piece	Echiro Oda	English	1100
7	Novel	Shirishko Phool	Parijat	Nepali	510
8	Sci-Fi	The Time Machine	H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Harayeko M...	Hari Bansha Acharya	Nepali	900
12	Fiction	Harry Potter	J.K. Rowling	English	1500

Figure 10:Before clicking clear button



Book No:

Genre: ----------------

Book Name:

Author: ----------------

Language: ○ English ○ Nepali

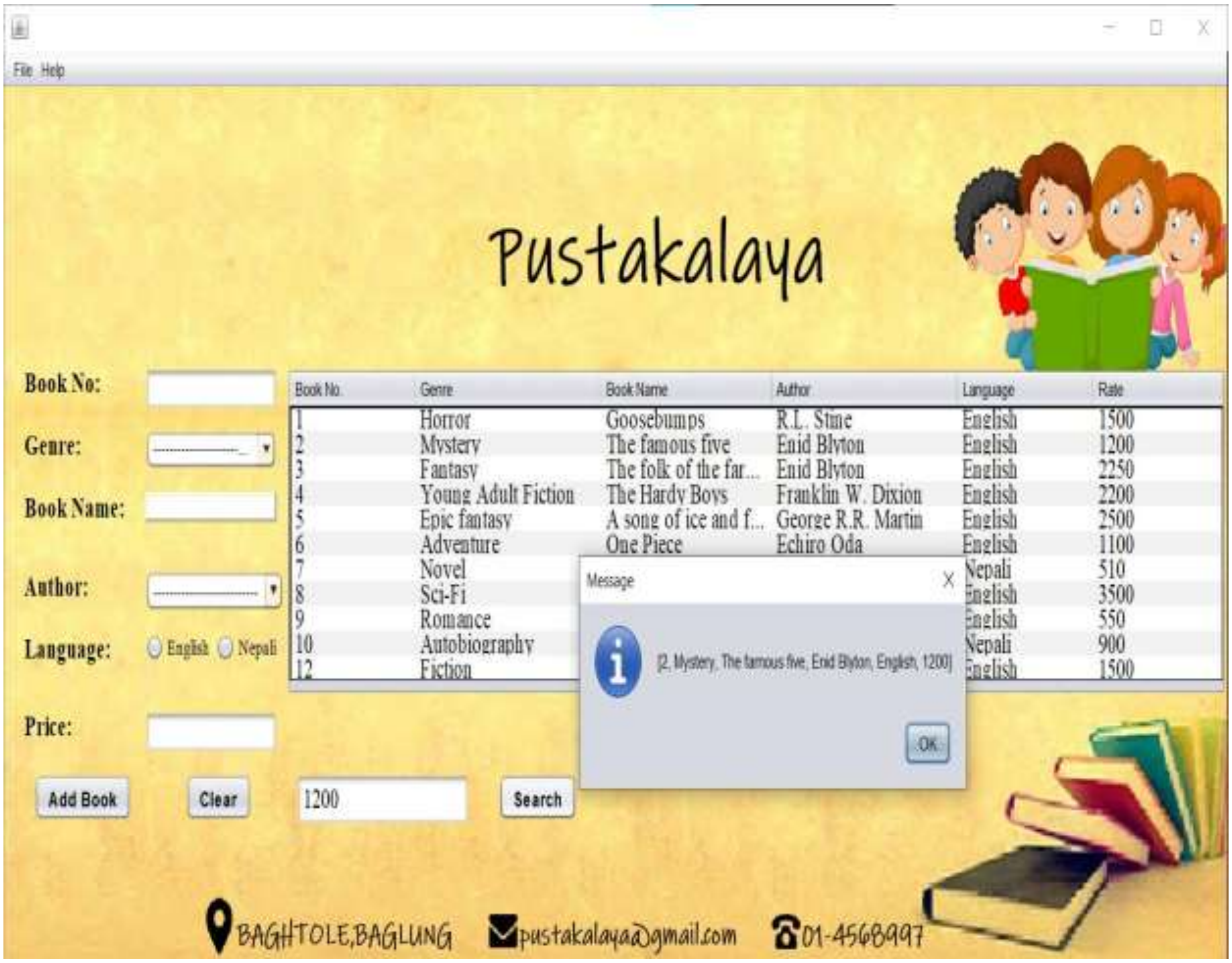
Price:

Book No	Genre	Book Name	Author	Language	Rate
1	Horror	Goosebumps	R.L. Stine	English	1500
2	Mystery	The famous five	Enid Blyton	English	1200
3	Fantasy	The folk of the far...	Enid Blyton	English	2250
4	Young Adult Fiction	The Hardy Boys	Franklin W. Dixion	English	2200
5	Epic fantasy	A song of ice and f...	George R.R. Martin	English	2500
6	Adventure	One Piece	Echiro Oda	English	1100
7	Novel	Shirishko Phool	Parijat	Nepali	510
8	Sci-Fi	The Time Machine	H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Harayeko M...	Hari Bansha Acharya	Nepali	900
12	Fiction	Harry Potter	J.K. Rowling	English	1500

Figure 11:After clicking clear button

7.4 Test 4: Searching book details by book price

Objective	➢ **To search the book details in by price.**
Action	➢ **At first, the added actual price of the book is entered** ➢ **Then the search button is clicked.**
Expected Result	➢ **The dialog box must be appeared and matching details of the book by price must be shown.**
Actual Result	➢ **The dialogue box with matching items by price is displayed.**
Conclusion	➢ **The test was carried out successfully.**



1200

Book No.	Genre	Book Name	Author	Language	Rate
1	Horror	Goosebumps	R.L. Stine	English	1500
2	Mystery	The famous five	Enid Blyton	English	1200
3	Fantasy	The folk of the far...	Enid Blyton	English	2250
4	Young Adult Fiction	The Hardy Boys	Franklin W. Dixion	English	2200
5	Epic fantasy	A song of ice and f...	George R.R. Martin	English	2500
6	Adventure	One Piece	Echiro Oda	English	1100
7	Novel			Nepali	510
8	Sci-Fi			English	3500
9	Romance			English	550
10	Autobiography			Nepali	900
12	Fiction			English	1500

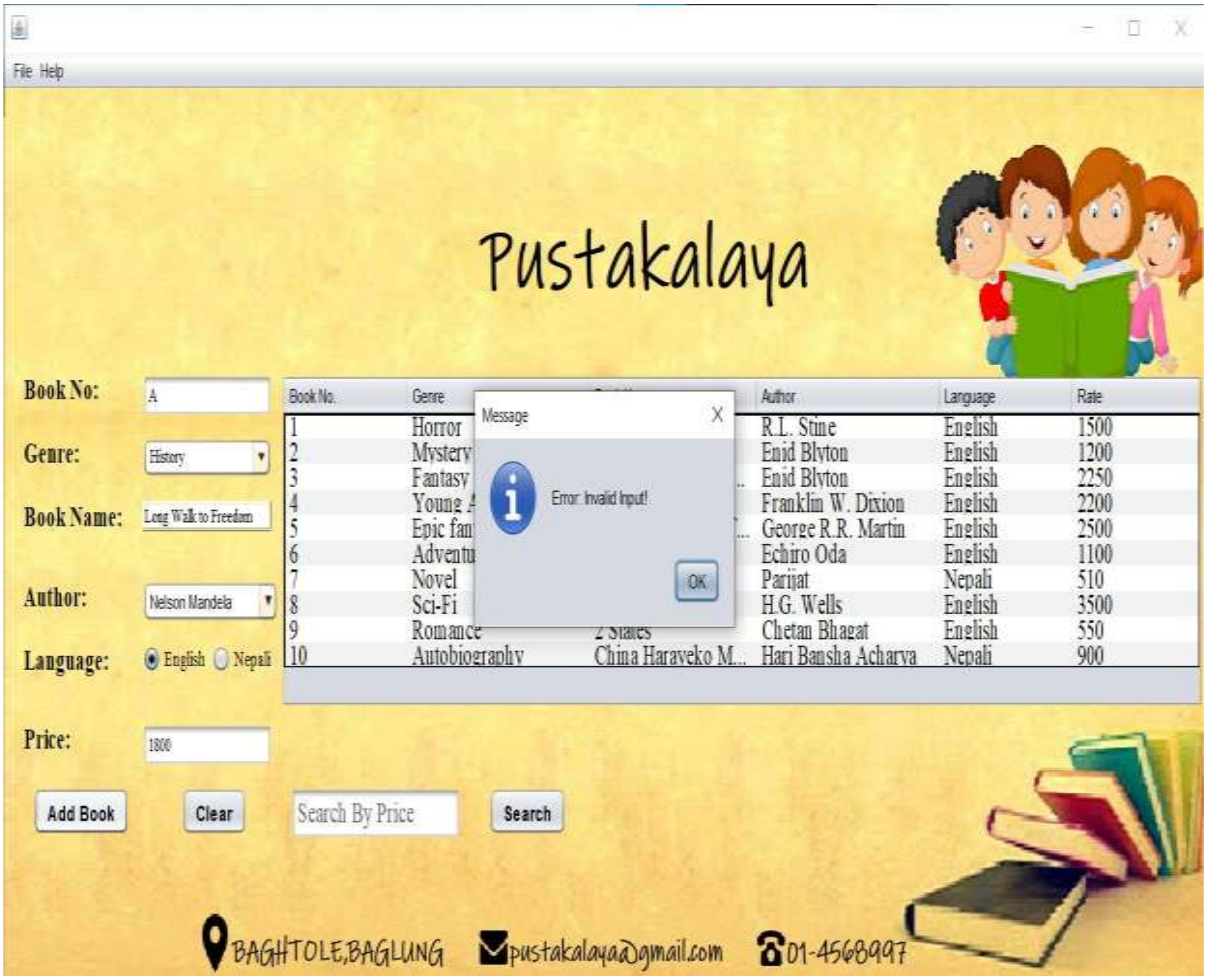
[2, Mystery, The famous five, Enid Blyton, English, 1200]

Figure 12:Result of test 4

7.5 Test 5 : Handling Errors

Test 5.1:Passing non-unique ID

Objective	➢ **To check the input validation**
Action	➢ **Non – unique ID was given as an Input in the Book ID** ➢ **Then add button is clicked**
Expected Result	➢ **A dialog box with an error message should appear**
Actual Result	➢ **Dialog box with an error message appeared**
Conclusion	➢ **The test was carried out successfully.**



Book No: A

Genre: History

Book Name: Long Walk to Freedom

Author: Nelson Mandela

Language: ● English ○ Nepali

Price: 1800

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy	...	Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan	...	George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Haraveko M...	Hari Bansha Acharya	Nepali	900

Message

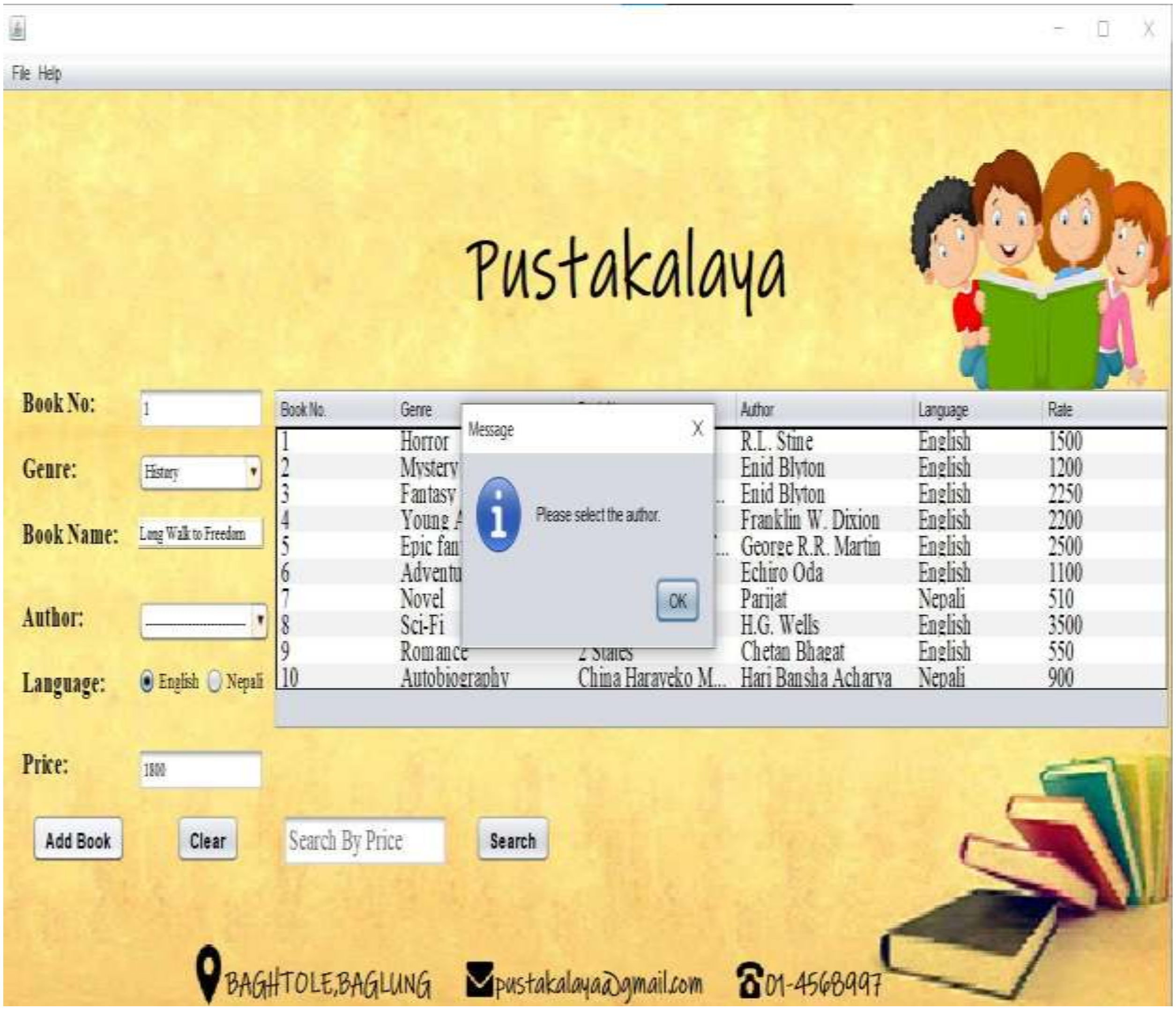
Error: Invalid Input!

OK

Figure 13:Result of test 5.1

Test 5.2 : Leaving author name empty

Objective	➢ **To check the input validation**
Action	➢ **No input was given in author name** ➢ **Then the add button is clicked**
Expected Result	➢ **A dialog box with an error message should appear**
Actual Result	➢ **Dialog box with an error message appeared**
Conclusion	➢ **The test was carried out successfully.**



Book No: 1

Genre: History

Book Name: Long Walk to Freedom

Author: ------------------

Language: ● English ○ Nepali

Price: 1800

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy	...	Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan	...	George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Harayeko M...	Hari Bansha Acharya	Nepali	900

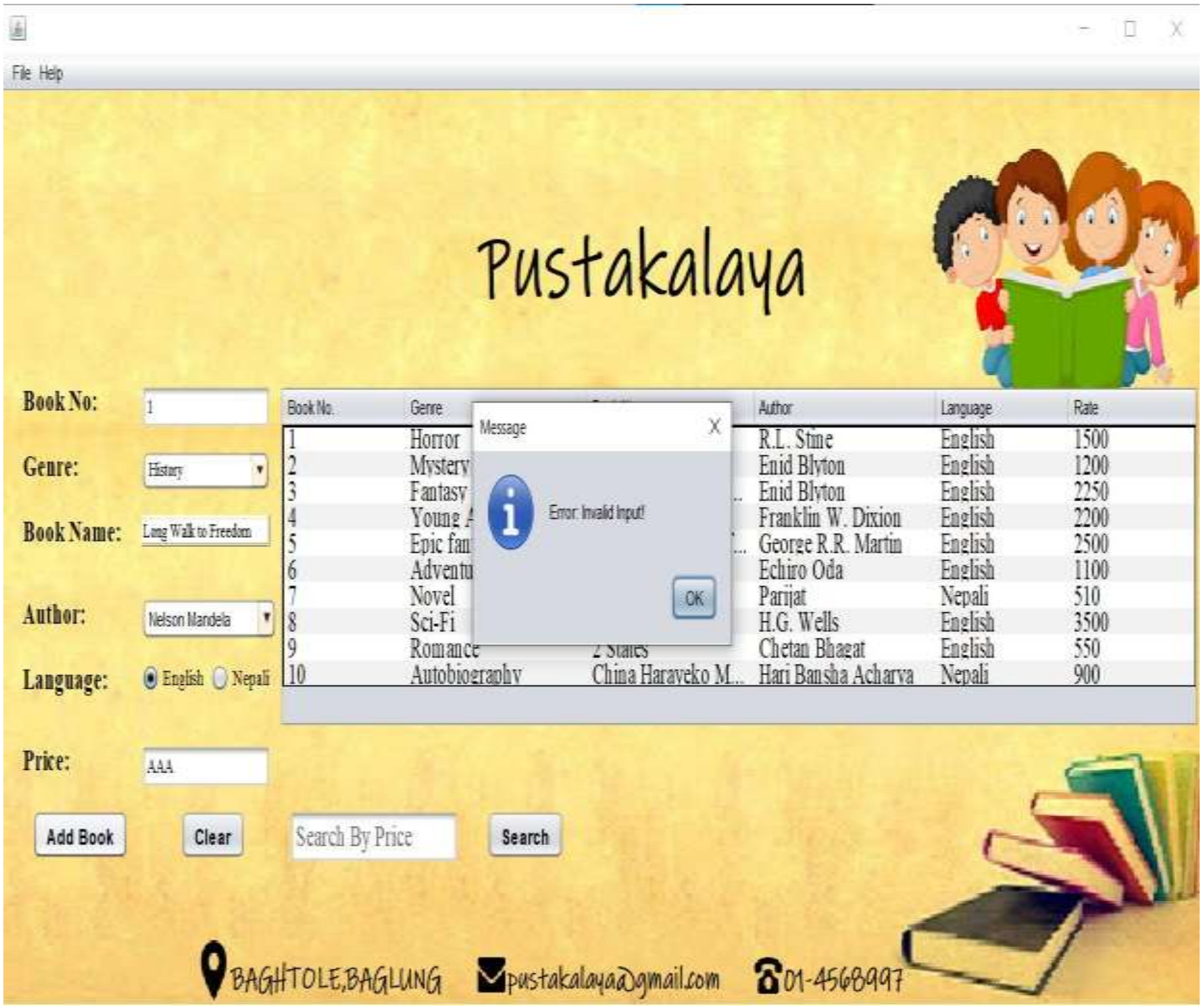
Message

Please select the author.

Figure 14:Result of test 5.2

Test 5.3 : Passing String in Price

Objective	➢ **To check the input validation**
Action	➢ **String is given as an input in the price** ➢ **Then the add button is clicked**
Expected Result	➢ **A dialog box with an error message should appear**
Actual Result	➢ **Dialog box with an error message appeared**
Conclusion	➢ **The test was carried out successfully.**



File Help

Pustakalaya

Book No: 1

Genre: History

Book Name: Long Walk to Freedom

Author: Nelson Mandela

Language: ● English ○ Nepali

Price: AAA

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy	..	Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan	...	George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Haraveko M...	Hari Bansha Acharya	Nepali	900

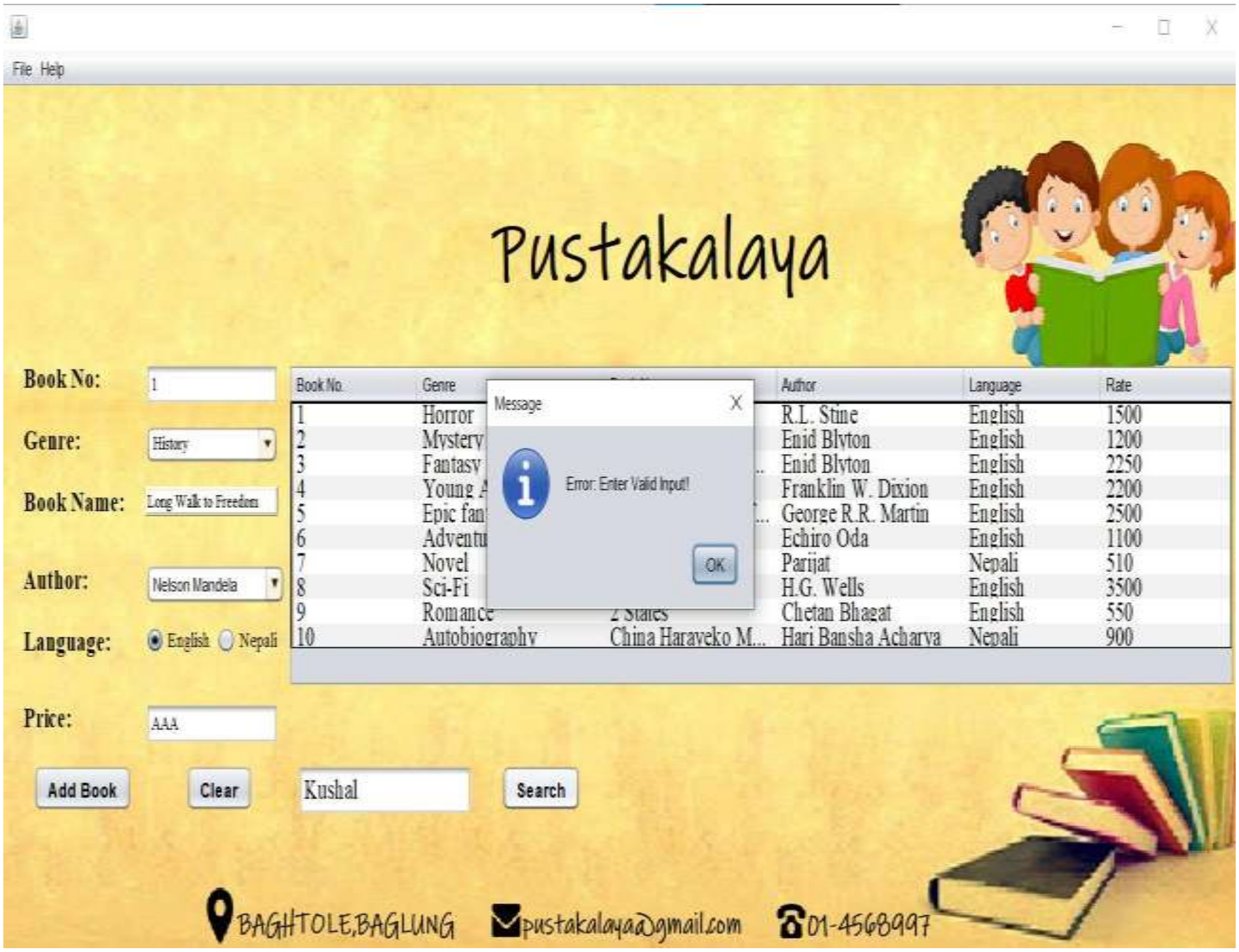
Message

Error: Invalid Input!

Figure 15:Result of test 5.3

Test 5.4: Not Passing value in search area

Objective	➢ **To check the input validation**
Action	➢ **Clicking Search button without giving any input in the search area**
Expected Result	➢ **A dialog box with an error message should appear**
Actual Result	➢ **Dialog box with an error message appeared**
Conclusion	➢ **The test was carried out successfully.**



Book No: 1

Genre: History

Book Name: Long Walk to Freedom

Author: Nelson Mandela

Language: ● English ○ Nepali

Price: AAA

Kushal

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy	...	Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan	...	George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Haraveko M...	Hari Bansha Acharya	Nepali	900

Message

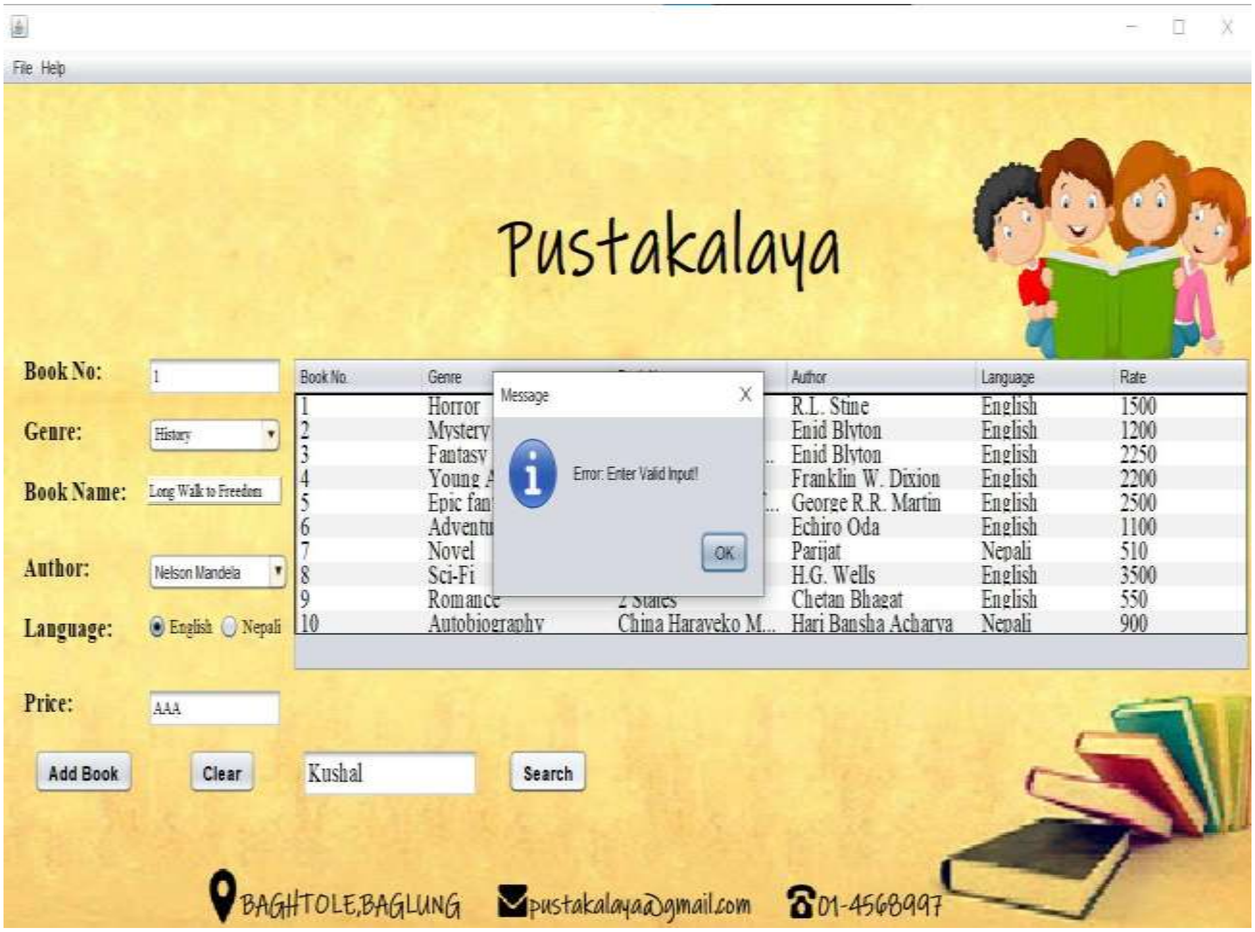
Error: Enter Valid Input!

OK

Figure 16:Result of test 5.4

Test 5.5: Passing unsuitable value in search area

Objective	➢ **To check the input validation**
Action	➢ **A string value is entered in the search area** ➢ **Then the search button is clicked**
Expected Result	➢ **A dialog box with an error message should appear**
Actual Result	➢ **Dialog box with an error message appeared**
Conclusion	➢ **The test was carried out successfully.**



Book No: 1

Genre: History

Book Name: Long Walk to Freedom

Author: Nelson Mandela

Language: ● English ○ Nepali

Price: AAA

Kushal

Message

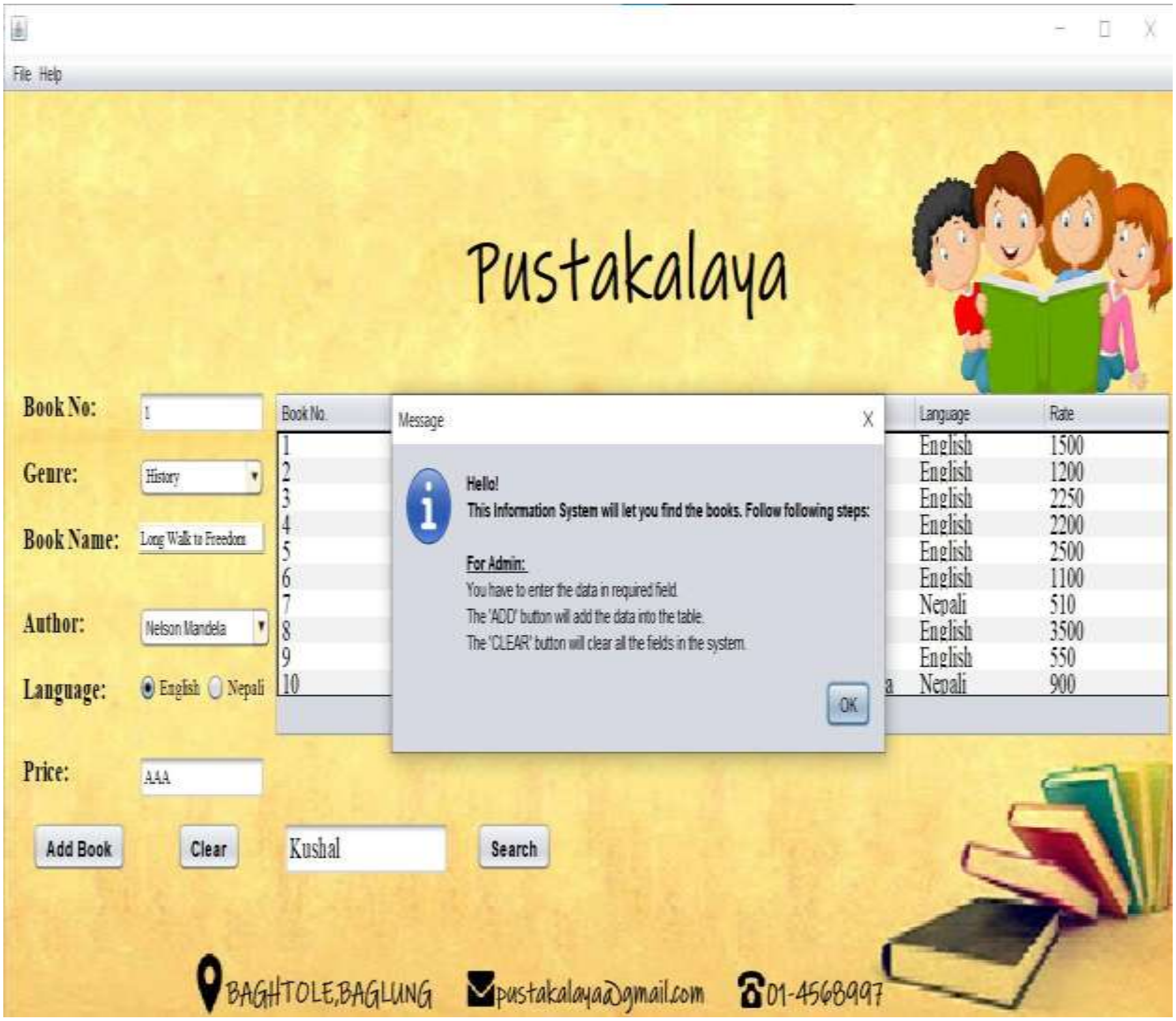
Error: Enter Valid Input!

Book No.	Genre		Author	Language	Rate
1	Horror		R.L. Stine	English	1500
2	Mystery		Enid Blyton	English	1200
3	Fantasy	...	Enid Blyton	English	2250
4	Young A		Franklin W. Dixion	English	2200
5	Epic fan	...	George R.R. Martin	English	2500
6	Adventu		Echiro Oda	English	1100
7	Novel		Parijat	Nepali	510
8	Sci-Fi		H.G. Wells	English	3500
9	Romance	2 States	Chetan Bhagat	English	550
10	Autobiography	China Haraveko M...	Hari Bansha Acharya	Nepali	900

Figure 17:Result of test 5.5

7.6 Test 6 : Help menu

Objective	➢ **To check the help menu**
Action	➢ **Help option from the menu bar is clicked**
Expected Result	➢ **A window with information should appear**
Actual Result	➢ **A window with information appeared**
Conclusion	➢ **The test was carried out successfully**



Book No: 1

Genre: History

Book Name: Long Walk to Freedom

Author: Nelson Mandela

Language: ● English ○ Nepali

Price: AAA

Kushal

Book No.		Language	Rate
1		English	1500
2		English	1200
3		English	2250
4		English	2200
5		English	2500
6		English	1100
7		Nepali	510
8		English	3500
9		English	550
10	a	Nepali	900

Message

Hello!

This Information System will let you find the books. Follow following steps:

For Admin:

You have to enter the data in required field.
The 'ADD' button will add the data into the table.
The 'CLEAR' button will clear all the fields in the system.

Figure 18:Result of test 6

8. Conclusion

This project is all about the coursework of our programming subject where we are given group assignment task of developing java program to create graphical user interface for inventory information system by implementing our knowledge that we gained in our tutorial classes. All learning stuff like what we learnt in our lecture and tutorial session that we have to implement in this project by developing a GUI based program for the bookstore information system.

To accomplish this project, at first we should understand about java programming more effectively and about knowledge of various methods such as parameterized constructor, Inheritance, uses of various keywords, accessor, mutator methods, control statement like if-else statement, variable, datatype and syntax errors, logical errors, action performed methods, validation, binary search algorithm, sorting, searching, NetBeans components and so on. The programming application is based on java programming and contains encapsulation, inheritance, and object casting and so on services. This group work is quite hard job for us especially the coding part. We did research on many important things that is required in our coding and we team members, together gave our 100% to complete our project. We have tried and tried thousand times to complete our coding without any errors and bugs. Thus, through the co-ordination of our team members and help of our tutor and teaching stuffs on the internet, we learnt and gained many things based on java programming that helps us to accomplish our goal on this project smoothly.

In this project, we used software named NetBeans to do java programming, coding. At the beginning of this project, we felt so many confusion while doing coding like how to deal with methods to do programming in a good program style then through continuous research on books, tutorials, lectures, teaching stuffs on internet and help from our teachers, we are able to cope with our confusion & obstacles. Even though various bugs such as Syntax errors, runtime errors, logical errors were encountered and simple mistakes such as not applying proper lower, upper cases and missing braces, semicolon and so on. We have given our continuous effort on this project, as we are the beginner person so; it consumed lots of time and patience to accomplish this project in a proper way.

With many relevant researches, studies on books, websites about java programming and assistance of our module teacher, we could understand the aim and objective of this project and cope with the doubts that raised while doing this coursework such as how to apply suitable methods for programming, how to do searching and sorting, how to do validation, how to do testing, how to make better and attractive design for GUI, how to do binary and linear search etc. Though many difficulties and obstacles were encountered, this coursework is easily completed through the help of our module teacher and relevant studies on our lectures slides.

With the assistance of this GUI based application, it will make any type of organization and inventory information system to store any kind of information in a reliable way. After using this application, no longer will the bookstore information system face difficulties to store the information of any type of books with its respective book genre, price, language etc. here the GUI which we developed stores each and every required part. Thus, the main goal of this program is to create a GUI program that helps to store details of Microsoft Product and licensed issued details in the list.

From this project, we have learnt many things logically and practically also. Several importances of java programming and positive conclusion are achieved from this program application practically in real life and in different organizations too. We must say that this coursework will be very useful for us in our future too because the skill and knowledge that we grabbed from developing GUI for the bookstore information system might be used further by us in GUI development for any other project in future too.

This group work project could not have been accomplished without the effort, support, patience, and co-operation of our group members (Surakshya Shrestha, Pratigya Rana, Kushal Piya, and Rohin Bharatee). We are grateful that we met all the requirements of this coursework and accomplished it within the submission date. Finally, yet importantly, we acknowledge with admiration the sacrifices made by our family, teachers because of our involvement with the task of completing this report. Concisely, it was a great comfort, experience and relief completing this coursework.

9. References

GeeksforGeeks, 2020. *geeksforgeeks*. [Online] Available at: https://www.geeksforgeeks.org/binary-insertion-sort/ [Accessed 09 January 2021].
Lifewire, 2021. *Lifewire*. [Online] Available at: https://www.lifewire.com/microsoft-word-4159373 [Accessed 10 January 2021].
Oracle, 2021. *Oracle*. [Online] Available at: https://www.oracle.com/tools/technologies/netbeans-ide.html [Accessed 10 January 2021].
Ryder, A., 2019. *Open Genius IQ*. [Online] Available at: https://iq.opengenus.org/binary-insertion-sort/ [Accessed 14 January 2021].
techopedia, January 18, 2017. *What is a Binary Search_ - Definition from Techopedia.html*. [Online] Available at: https://www.techopedia.com/definition/14270/binary-search [Accessed 03 January 2021].

10. Appendix

```
import java.awt.Color;
import java.awt.HeadlessException;
import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author hp
 */
public class Bookinfo extends javax.swing.JFrame {

    DefaultTableModel tableModel;
    public ArrayList<table> addtotable = new ArrayList<>();

    public Bookinfo() {
        initComponents();
        tableModel = (DefaultTableModel) jTable1.getModel();
        dataadd();
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        languageRadioGroup = new javax.swing.ButtonGroup();
        jFileChooser1 = new javax.swing.JFileChooser();
        addBookBtn = new javax.swing.JButton();
        title = new javax.swing.JLabel();
        clearBtn = new javax.swing.JButton();
        searchText = new javax.swing.JTextField();
        searchBtn = new javax.swing.JButton();
        priceComboBox = new javax.swing.JTextField();
        price = new javax.swing.JLabel();
        language = new javax.swing.JLabel();
        englishRadioButton = new javax.swing.JRadioButton();
        nepaliRadioButton = new javax.swing.JRadioButton();
        authorComboBox = new javax.swing.JComboBox<>();
        bookNameText = new javax.swing.JTextField();
        genreComboBox = new javax.swing.JComboBox<>();
        bookNoText = new javax.swing.JTextField();
        genre = new javax.swing.JLabel();
        bookName = new javax.swing.JLabel();
        author = new javax.swing.JLabel();
        bookNo = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
```

```
location = new javax.swing.JLabel();
locationIcon = new javax.swing.JLabel();
email = new javax.swing.JLabel();
emailIcon = new javax.swing.JLabel();
phoneNumber = new javax.swing.JLabel();
phoneIcon = new javax.swing.JLabel();
children = new javax.swing.JLabel();
background = new javax.swing.JLabel();
jMenuBar2 = new javax.swing.JMenuBar();
jMenu5 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
jMenu6 = new javax.swing.JMenu();

jFileChooser1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jFileChooser1ActionPerformed(evt);
    }
});

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(204, 255, 51));
setSize(new java.awt.Dimension(500, 300));
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

addBookBtn.setFont(new java.awt.Font("SansSerif", 1, 14)); // NOI18N
addBookBtn.setText("Add Book");
addBookBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addBookBtnActionPerformed(evt);
    }
```

```
        });
        getContentPane().add(addBookBtn, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 440, -1, 30));

        title.setFont(new java.awt.Font("Ink Free", 1, 60)); // NOI18N
        title.setText("Pustakalaya");
        title.setToolTipText("");
        getContentPane().add(title, new
org.netbeans.lib.awtextra.AbsoluteConstraints(480, 70, 360, -1));

        clearBtn.setFont(new java.awt.Font("SansSerif", 1, 14)); // NOI18N
        clearBtn.setText("Clear");
        clearBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                clearBtnActionPerformed(evt);
            }
        });
        getContentPane().add(clearBtn, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180, 440, -1, 30));

        searchText.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
        searchText.setForeground(new java.awt.Color(102, 102, 102));
        searchText.setText("Search By Price");
        searchText.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusGained(java.awt.event.FocusEvent evt) {
                searchTextFocusGained(evt);
            }
            public void focusLost(java.awt.event.FocusEvent evt) {
                searchTextFocusLost(evt);
            }
        });
        searchText.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                searchTextActionPerformed(evt);
            }
        });
        getContentPane().add(searchText, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 440, 170, -1));

        searchBtn.setFont(new java.awt.Font("SansSerif", 1, 14)); // NOI18N
        searchBtn.setText("Search");
        searchBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                searchBtnActionPerformed(evt);
            }
        });
        getContentPane().add(searchBtn, new
org.netbeans.lib.awtextra.AbsoluteConstraints(490, 440, -1, 30));

        priceComboBox.setFont(new java.awt.Font("Times New Roman", 0, 12)); //
NOI18N
        priceComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                priceComboBoxActionPerformed(evt);
            }
        });
        getContentPane().add(priceComboBox, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 400, 130, -1));

        price.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        price.setText("Price:");
        getContentPane().add(price, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 400, -1, -1));

        language.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        language.setText("Language:");
        getContentPane().add(language, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 350, -1, -1));

        languageRadioGroup.add(englishRadioButton);
        englishRadioButton.setFont(new java.awt.Font("Times New Roman", 0, 14)); //
NOI18N
        englishRadioButton.setText("English");
        getContentPane().add(englishRadioButton, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, -1, -1));

        languageRadioGroup.add(nepaliRadioButton);
        nepaliRadioButton.setFont(new java.awt.Font("Times New Roman", 0, 14)); //
NOI18N
        nepaliRadioButton.setText("Nepali");
        getContentPane().add(nepaliRadioButton, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 350, -1, -1));

        authorComboBox.setBackground(new java.awt.Color(255, 204, 102));
        authorComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "--------------------------", "J.K. Rowling", "Arthur Conan Doyle", "Amish Tripathi",
"Rahul Badami", "Sameer Kamat", "Nelson Mandela" }));
        getContentPane().add(authorComboBox, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 310, -1, -1));

        bookNameText.setFont(new java.awt.Font("Times New Roman", 0, 12)); //
NOI18N
```

```
bookNameText.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bo
rder.BevelBorder.RAISED));
        bookNameText.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bookNameTextActionPerformed(evt);
            }
        });
        getContentPane().add(bookNameText, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 260, 130, -1));

        genreComboBox.setBackground(new java.awt.Color(255, 204, 102));
        genreComboBox.setFont(new java.awt.Font("Times New Roman", 0, 12)); //
NOI18N
        genreComboBox.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "--------------------------", "Action and Adventure", "History", "Fiction", "Non-
Fiction", "Romance" }));
        genreComboBox.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                genreComboBoxActionPerformed(evt);
            }
        });
        getContentPane().add(genreComboBox, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 220, 130, -1));

        bookNoText.setFont(new java.awt.Font("Times New Roman", 0, 12)); // NOI18N
        bookNoText.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                bookNoTextActionPerformed(evt);
            }
        });
        getContentPane().add(bookNoText, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 180, 130, -1));

        genre.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        genre.setText("Genre:");
        getContentPane().add(genre, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 220, -1, -1));

        bookName.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        bookName.setText("Book Name:");
        getContentPane().add(bookName, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 260, -1, -1));

        author.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        author.setText("Author:");
        getContentPane().add(author, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, -1, -1));

        bookNo.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        bookNo.setText("Book No:");
        getContentPane().add(bookNo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 180, -1, -1));

        jTable1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
        jTable1.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
```

```
            "Book No.", "Genre", "Book Name", "Author", "Language", "Rate"
          }
        ));
        jScrollPane1.setViewportView(jTable1);
        if (jTable1.getColumnModel().getColumnCount() > 0) {
          jTable1.getColumnModel().getColumn(0).setPreferredWidth(5);
          jTable1.getColumnModel().getColumn(2).setPreferredWidth(60);
          jTable1.getColumnModel().getColumn(4).setPreferredWidth(25);
          jTable1.getColumnModel().getColumn(5).setPreferredWidth(20);
        }

        getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(280, 180, 930, 210));

        location.setFont(new java.awt.Font("Ink Free", 1, 20)); // NOI18N
        location.setText("BAGHTOLE,BAGLUNG");
        getContentPane().add(location, new
org.netbeans.lib.awtextra.AbsoluteConstraints(230, 530, -1, -1));

        locationIcon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/location.png"))); // NOI18N
        getContentPane().add(locationIcon, new
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 520, -1, -1));

        email.setFont(new java.awt.Font("Ink Free", 1, 20)); // NOI18N
        email.setText("pustakalaya@gmail.com");
        getContentPane().add(email, new
org.netbeans.lib.awtextra.AbsoluteConstraints(510, 530, -1, -1));

        emailIcon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/mail.png"))); // NOI18N
```

```
        getContentPane().add(emailIcon, new
org.netbeans.lib.awtextra.AbsoluteConstraints(480, 530, -1, -1));

        phoneNumber.setFont(new java.awt.Font("Ink Free", 1, 20)); // NOI18N
        phoneNumber.setText("01-4568997");
        getContentPane().add(phoneNumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(790, 530, -1, -1));

        phoneIcon.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/phone.png"))); // NOI18N
        getContentPane().add(phoneIcon, new
org.netbeans.lib.awtextra.AbsoluteConstraints(760, 530, -1, -1));

        children.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/children.png"))); // NOI18N
        getContentPane().add(children, new
org.netbeans.lib.awtextra.AbsoluteConstraints(910, 10, -1, -1));

        background.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/background.jpg"))); // NOI18N
        getContentPane().add(background, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        jMenu5.setText("File");

        jMenuItem1.setText("Open");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem1);

        jMenuItem2.setText("Exit");
        jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem2ActionPerformed(evt);
            }
        });
        jMenu5.add(jMenuItem2);

        jMenuBar2.add(jMenu5);

        jMenu6.setText("Help");
        jMenu6.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jMenu6MouseClicked(evt);
            }
        });
        jMenu6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu6ActionPerformed(evt);
            }
        });
        jMenuBar2.add(jMenu6);

        setJMenuBar(jMenuBar2);

        pack();
    }// </editor-fold>

    public boolean noDuplicateData(String productModel, int index) {
        boolean noDup = false;
        int rows = jTable1.getRowCount();
        int count = 0;
        for (int j = 0; j < rows; j++) {
            if (String.valueOf(jTable1.getValueAt(j, index)).equals(productModel)) {
                count++;
            }
        }
        if (count == 0) {
            noDup = true;
            System.out.println(noDup);
        }
        return noDup;
    }


    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String searchValue = searchText.getText();
        int rowCount = jTable1.getModel().getRowCount();
        int colCount = jTable1.getModel().getColumnCount();
        System.out.print("ROWCOUNT" + rowCount);

        int nextRow = 0;
        boolean emptyRow = false;
        try {
            if (rowCount != 0) {
```

```
do {
    if (jTable1.getValueAt(nextRow, 0) != null) {
        nextRow++;

    } else {
        emptyRow = true;
    }
} while (nextRow < rowCount && !emptyRow);
System.out.println("NextRow " + nextRow);

int price[] = new int[nextRow];
System.out.println(price.length);
for (int i = 0; i < nextRow; i++) {
    String value = jTable1.getModel().getValueAt(i, 5).toString();
    System.out.println("String " + value);
    int valNum = Integer.parseInt(value);
    price[i] = valNum;
    System.out.print(Arrays.toString(price));
}
System.out.println(Arrays.toString(price));
sort(price);
System.out.println(Arrays.toString(price));
int low = 0;
int high = price.length;
int valueInt = Integer.parseInt(searchValue);
int search = numberSearch(price, low, high, valueInt);
System.out.print(" ");
if (search != -1) {
    String val = String.valueOf(price[search]);
    System.out.print("here 1:" + val);
    String searchArray[] = new String[6];
```

```
                    int rowNext = 0;
                    int rowSearch = 0;
                    boolean emptyCheckPrice = false;

                    if (rowCount != 0) {
                        do {
                            if (jTable1.getValueAt(rowNext, 0) != null) {
                                System.out.println(rowNext);
                                String valTable = jTable1.getModel().getValueAt(rowNext,
5).toString();
                                System.out.println("from table" + valTable);
                                if (val.equals(valTable)) {
                                    for (int i = 0; i < colCount; i++) {
                                        searchArray[i] = jTable1.getModel().getValueAt(rowNext,
i).toString();
                                    }
                                    System.out.println("from array" + val);
                                    System.out.println(Arrays.toString(searchArray));
                                    JOptionPane.showMessageDialog(null,
Arrays.toString(searchArray));
                                    rowSearch++;

                                }
                                rowNext++;
                            } else {
                                emptyCheckPrice = true;
                            }
                        } while (rowNext < rowCount && !emptyCheckPrice);
                    }
                    System.out.println(rowNext);
                } else {
                    JOptionPane.showMessageDialog(rootPane, "No entry found.");
                }
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, "Error: Enter Valid Input!");
        }
    }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFileChooser open = new JFileChooser();
        open.showOpenDialog(null);
        File f = open.getSelectedFile();
    }

    private void searchTextActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jMenu6ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jMenu6MouseClicked(java.awt.event.MouseEvent evt) {
        JOptionPane.showMessageDialog(null, "<html><b>Hello! <br>"
                + "This Information System will let you find the books. Follow following steps:
<br>"
                + "<br>"
                + "<u>For Admin: </u> </b> <br>"
                + "You have to enter the data in required field. <br>"
                + "The 'ADD' button will add the data into the table. <br>"
                + "The 'CLEAR' button will clear all the fields in the system. <br>");

    }

    private void searchTextFocusLost(java.awt.event.FocusEvent evt) {
        if (searchText.getText().equals("")) {
            searchText.setText("Search By Price");
            searchText.setForeground(new Color(102, 102, 102, 102));
        }
    }

    private void searchTextFocusGained(java.awt.event.FocusEvent evt) {
        if (searchText.getText().equals("Search By Price")) {
            searchText.setText("");
            searchText.setForeground(new Color(0, 0, 0));
        }
    }

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

        try {
            int bookid = Integer.parseInt(bookNoText.getText());
            String genre = (String) genreComboBox.getSelectedItem();
            String bookname = bookNameText.getText();
            String author = (String) authorComboBox.getSelectedItem();
            String language = null;
            if (englishRadioButton.isSelected()) {
                language = englishRadioButton.getText();
            } else if (nepaliRadioButton.isSelected()) {
                language = nepaliRadioButton.getText();
            }
            int rate = Integer.parseInt(priceComboBox.getText());
            table tab = new table(bookid, genre, bookname, author, language, rate);
            if (bookid > 0) {
                if (noDuplicateData(bookname, 2)) {
                    if (genreComboBox.getSelectedIndex() != 0) {
                        if (!bookname.isEmpty()) {
                            if (authorComboBox.getSelectedIndex() != 0) {
                                if (language != null) {
                                    if (rate > 0) {
                                        addtotable.add(tab);
                                        tableModel.addRow(tab.getbook());
                                    } else {
                                        JOptionPane.showMessageDialog(rootPane, "Please input the
rate");
                                    }
                                } else {
                                    JOptionPane.showMessageDialog(rootPane, "Please select the
language");
                                }
                            } else {
                                JOptionPane.showMessageDialog(rootPane, "Please select the
author.");
                            }
                        } else {
                            JOptionPane.showMessageDialog(rootPane, "Please enter the book
name.");
                        }
                    } else {
                        JOptionPane.showMessageDialog(rootPane, "Please select the genre.");
                    }
                } else {
```

```
                JOptionPane.showMessageDialog(rootPane, "Duplicate Entries Found!");
            }
        } else {
            JOptionPane.showMessageDialog(rootPane, "Please input the bookid");
        }
    } catch (HeadlessException | NumberFormatException e) {
        JOptionPane.showMessageDialog(rootPane, "Error: Invalid Input!");
    }
}

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


}

private void bookNameTextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {
    bookNoText.setText("");
    genreComboBox.setSelectedIndex(0);
    bookNameText.setText("");
    authorComboBox.setSelectedIndex(0);
    languageRadioGroup.clearSelection();
    priceComboBox.setText("");
}

private void bookNoTextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

```
    private void genreComboBoxActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void jFileChooser1ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(Bookinfo.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Bookinfo.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Bookinfo.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Bookinfo.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Bookinfo().setVisible(true);
            }
        });
    }

    public void dataadd() {
        table b1 = new table(1, "Horror", "Goosebumps", "R.L. Stine", "English", 1500);
        table b2 = new table(2, "Mystery", "The famous five", "Enid Blyton", "English",
1200);
```

```
        table b3 = new table(3, "Fantasy", "The folk of the faraway tree", "Enid Blyton", "English", 2250);
        table b4 = new table(4, "Young Adult Fiction", "The Hardy Boys", "Franklin W. Dixion", "English", 2200);
        table b5 = new table(5, "Epic fantasy", "A song of ice and fire", "George R.R. Martin", "English", 2500);
        table b6 = new table(6, "Adventure", "One Piece", "Echiro Oda", "English", 1100);
        table b7 = new table(7, "Novel", "Shirishko Phool", "Parijat", "Nepali", 510);
        table b8 = new table(8, "Sci-Fi", "The Time Machine", "H.G. Wells", "English", 3500);
        table b9 = new table(9, "Romance", "2 States", "Chetan Bhagat", "English", 550);
        table b10 = new table(10, "Autobiography", "China Harayeko Manchhe", "Hari Bansha Acharya", "Nepali", 900);

        tableModel.addRow(b1.getbook());
        tableModel.addRow(b2.getbook());
        tableModel.addRow(b3.getbook());
        tableModel.addRow(b4.getbook());
        tableModel.addRow(b5.getbook());
        tableModel.addRow(b6.getbook());
        tableModel.addRow(b7.getbook());
        tableModel.addRow(b8.getbook());
        tableModel.addRow(b9.getbook());
        tableModel.addRow(b10.getbook());

        addtotable.add(b1);
        addtotable.add(b2);
        addtotable.add(b3);
        addtotable.add(b4);
        addtotable.add(b5);
        addtotable.add(b6);
        addtotable.add(b7);
        addtotable.add(b8);
        addtotable.add(b9);
        addtotable.add(b10);
    }

    public int numberSearch(int[] a, int low, int high, int value) {
        if (low <= high) {
            try {
                int mid = (low + high) / 2;
                if (a[mid] == value) {
                    return mid;
                } else if (a[mid] < value) {
                    System.out.print("mid " + mid);
                    return numberSearch(a, mid + 1, high, value);
                } else {
                    System.out.print("mid " + mid);
                    return numberSearch(a, low, mid - 1, value);
                }
            } catch (Exception e) {
                return -1;
            }
        } else {
            return -1;
        }
    }

    public static void swap(int[] a, int i, int j) {
        int temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }

    public static void sort(int[] a) {
        for (int i = 0; i < a.length - 1; i++) {
            int minPos = minimumPosition(a, i);
            Bookinfo.swap(a, minPos, i);
        }
    }

    public static int minimumPosition(int[] a, int from) {
        int minPos = from;
        for (int i = from + 1; i < a.length; i++) {
            if (a[i] < a[minPos]) {
                minPos = i;
            }
        }
        return minPos;

    }

    public static int searchString(String[] arr, String x) {
        try {
            int l = 0, r = arr.length - 1;
            while (l <= r) {
                int m = l + (r - 1) / 2;
                int res = x.compareTo(arr[m]);
                if (res == 0) {
                    return m;
                } else if (res > 0) {
                    l = m + 1;
                } else if (res < 0) {
```

```
                r = m - 1;
            }
        }
    } catch (Exception e) {

        return -1;
    }
    return - 1;
}

// Variables declaration - do not modify
private javax.swing.JButton addBookBtn;
private javax.swing.JLabel author;
private javax.swing.JComboBox<String> authorComboBox;
private javax.swing.JLabel background;
private javax.swing.JLabel bookName;
private javax.swing.JTextField bookNameText;
private javax.swing.JLabel bookNo;
private javax.swing.JTextField bookNoText;
private javax.swing.JLabel children;
private javax.swing.JButton clearBtn;
private javax.swing.JLabel email;
private javax.swing.JLabel emailIcon;
private javax.swing.JRadioButton englishRadioButton;
private javax.swing.JLabel genre;
private javax.swing.JComboBox<String> genreComboBox;
private javax.swing.JFileChooser jFileChooser1;
private javax.swing.JMenu jMenu5;
private javax.swing.JMenu jMenu6;
private javax.swing.JMenuBar jMenuBar2;
private javax.swing.JMenuItem jMenuItem1;
```

```
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JLabel language;
    private javax.swing.ButtonGroup languageRadioGroup;
    private javax.swing.JLabel location;
    private javax.swing.JLabel locationIcon;
    private javax.swing.JRadioButton nepaliRadioButton;
    private javax.swing.JLabel phoneIcon;
    private javax.swing.JLabel phoneNumber;
    private javax.swing.JLabel price;
    private javax.swing.JTextField priceComboBox;
    private javax.swing.JButton searchBtn;
    private javax.swing.JTextField searchText;
    private javax.swing.JLabel title;
    // End of variables declaration
}
import bookinfo.table;
```