LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CS5004NI Emerging Programming Platforms & Technologies

30% Group Coursework

2018-19 autumn

Module Leader: Dhruva Sen

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1.	**Deepika Kattel**	**NP01CP4A170036**	**17030988**
2.	**Ashutosh Chauhan**	**NP01CP4A170032**	**17030976**
3.	**Bivisha Karki**	**NP01CP4A170050**	**17031028**

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

It is always a pleasure to remind the fine people around us for their direction and support which we have acknowledged to sustain our project as well as our skills.

We would like to express our sincere gratitude to our tutor Mr. Shreyash Mool for providing his invaluable guidance, comments and suggestions throughout the course of the project. We are glad to have him as our tutor because he constantly motivated us to work harder. Our lecturer Mr. Monil Adhikari was not less than our tutor as he encouraged us to make our project better in best possible way. Similarly, we would like thank Islington College, for providing us with project which enhanced our knowledge and rational capability to higher level. The project assigned to us is an opportunity to experiment our knowledge and learn something new.

Finally, we would like to thank our family and friends for generous encouragement, enthusiasm and invaluable assistance to us. Without all this, we might not be able to complete this project properly. And to our group, we really thank each other for not giving up in any moment and investing all of their time and effort until the project is concluded successfully.

Proposal:

The project that we are to develop here in this module is for a Bike retail store namely "Moto Racers" which offers a variety of bikes with different features. The program developed here is used to store the data of bikes available in the store. The program is also used for displaying the latest arrivals in our collection. This program is very user friendly and easy to handle. The following are the list of data and features that we have used in this program. The data and features stated below will help clarify the users about this program.

- **List of data:**

1. **Model No:**

This is a 'String' value which is used to specify the model number of the bike. It is a unique identification of each bike and its input is given by the help of text field.

2. **Model Name:**

This is a 'String' value which is used to specify the name of the bike. The input is given by the help of text field.

3. **Displacement:**

This is a 'String' value which is used to specify the displacement of the engine. The input is given by the help of combo box.

4. **Cooling System:**

This is a 'String' value which is used to specify the method of cooling the uses. The input is given by the help of radio button.

5. **Company:**

This is a 'String' value which is used to specify the company of the bike. The input is given by help of combo box.

6. **Price:**

This is an ‘Integer’ value which is used to specify the price of the bike. The input is given by the help of text field.

- **List of features:**

1. Simple and easy user interface.
2. The program has a feature to add any bike specification on a table.
3. The user can search about bikes according to the company name.
4. The user can search bikes according to price.
5. The user can delete any bike specification from the data stored in the table.

- **Tools Used**

We have chosen to use Java programming language for the development of this project. Java makes GUI (Graphical user interface) building much simpler and easier due to its Swing and AWT (Abstract Window Toolkit) package. We have chosen NetBeans as IDE (Integrated development environment) as the user interface is much more friendly and easier than other IDEs available. Also, the GUI builder present in this IDE is very unsophisticated which makes the development of this application a simple task.

Individual Task

Each member of our group has equal contribution in this project. Each of us have helped each other while development of this project. We have worked together as a team in both the programming section as well as the documentation part. Below we have specified the main parts that each of us individually carried in. The overall programming and documentation part have been completed by the equal contribution of all the members of our group.

Members name	**Tasks performed**
Deepika Kattel	1. GUI 2. Add function in the program 3. Help-User guide 4. Report writing
Ashutosh Chauhan	1. Update function 2. Binary search implementation 3. File open function 4. Report Writing
Bivisha Karki	1. Delete function 2. Filter by company 3. Report writing

Table of Contents:

Table of Figures:

Table of Tables:

1. Introduction

The purpose of this group work was to develop an information system including all the specified criteria's in the question. The criteria were to store and display all the items in the system, display meaningful message if error input is entered in text field, if no any data is selected from drop down, if no any data is selected from radio button, and search method executed on the basis of binary search algorithm.

This coursework assigned to us was comprised of all the things that were taught to us in our classes which made the completion of the coursework a bit easier. Appreciating the durability of the coursework, we did this project with great determination and effort to complete it before the allocated deadline. We divided certain portions of the coursework among each other which made our work easier and quicker.

In order to achieve the objectives of this project, we went through different books, websites, journals etc. With the help of research, we solved all the problems gradually and made this project a huge success. The success of the project was only possible because of co-ordination between our team members, proper understanding between each other and our continuous hard work. At the end, our hard work paid off as we completed our task before the deadline and learnt about a lot of things which we were unknown about previously.

2. Aims and Objective:

The main aims of this project are:

1. To carry out all the tasks given in the coursework within the given period of time
2. To complete all the individual and group tasks of the coursework in the best possible manner.
3. To accomplish the objective of the assigned project with team effort.

The main objectives of this project are:

1. Analyzing and implementing the case scenario assigned to us in an appropriate manner.
2. Comprising our knowledge, skill and ability in an elaborative way.
3. Implementing things imparted to us in a significant task by proper planning between the team fellows.
4. Accomplishing the coursework by handling pressure and managing time of all the team members.
5. Working as a team and forming a best squad performance.

3. Binary Search Algorithm

The most popular searching algorithms in java are linear and binary search algorithms. Linear search is very simple, to check if an element is present in the given list we compare it with every element in the list. If it is found then we print the location at which it occurs, otherwise the list doesn't contain the element we are searching.

Binary search algorithm is a search algorithm that finds the position of a target value within a sorted array. It compares the target value to the middle element of the array. It delivers better performance than sequential search because it starts with a collection whose elements are already sorted. It divides the sorted collection in half until the sought for item is found, or until it is determined that the item does not exist in the collection. (tutorialspoint, 2019)

In this coursework, we have implemented binary search algorithm in searching the bikes using the price of the bike by sorting all the data using the merge sort algorithm. Here, all the rows are extracted from the JTable and each row is made a sub-array inside an array which is basically a 2D array. Now, the whole 2D array is sorted using the merge sort algorithm

Merge sort is basically a **Divide and Conquer** algorithm to sort a given set of elements recursively and then merge them. This algorithm runs in O(n*log n) time in all cases, where n is the number of comparisons done. There are two major steps involved in merge sort, they are:

- Divide the unsorted array into n partitions, each partition contains 1 element. Here the one element is considered as sorted.
- Repeatedly merge partitioned units to produce new sub lists until there is only 1 sub list remaining. This will be the sorted list at the end (java2novice, 2019)

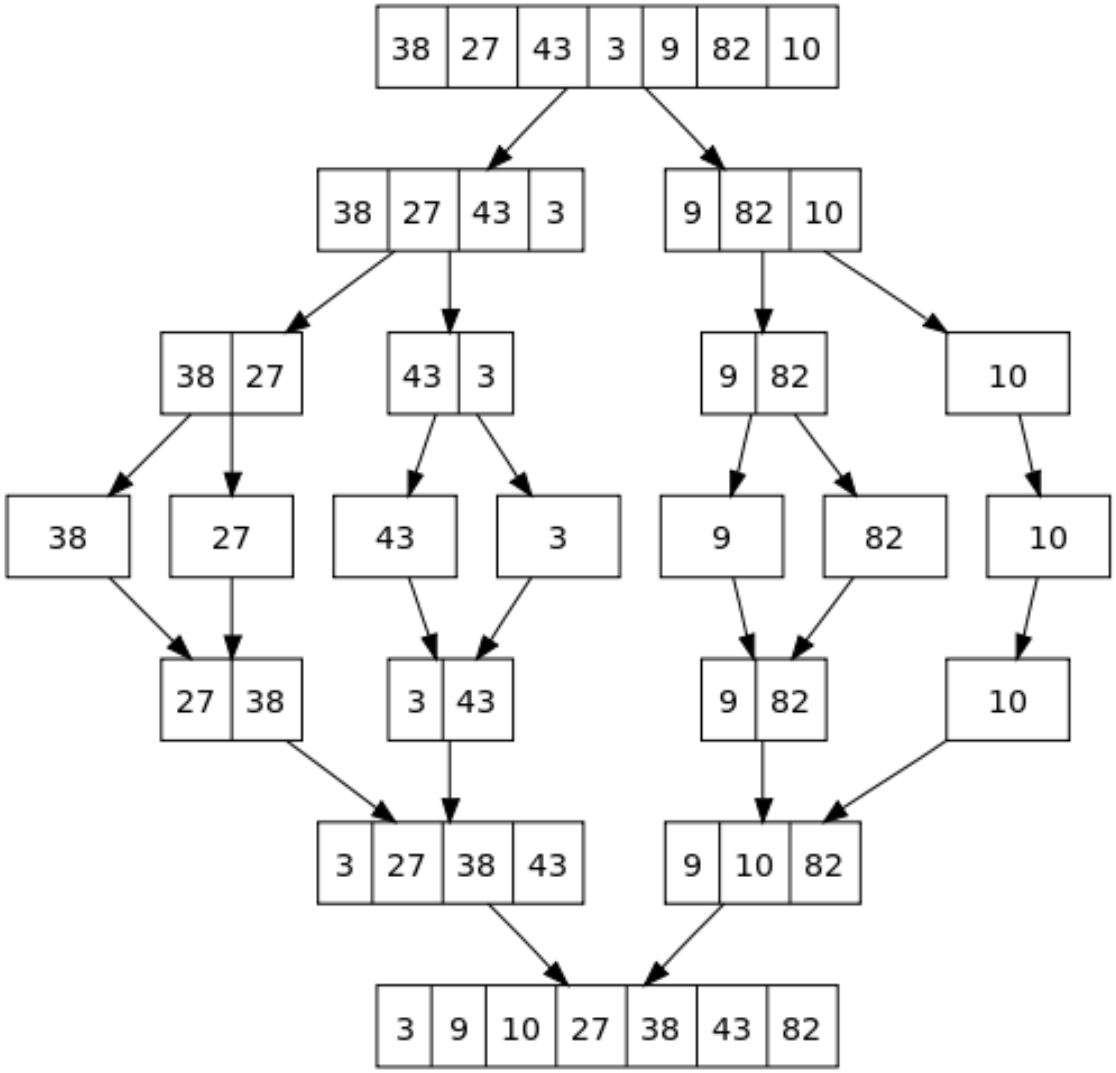


Figure 1: merge sorting an array (java2novice, 2019)

In the above figure, there is an array {38,27,43,3,9,82,10} which need to sorted. Here initially, the array is split into half and this process continues until each sub array contains only one element. Here, the only element remaining in the sub-array is considered sorted. Now, the single element array is compared to the array beside it, and merged them to produce a new sorted array until there is only one array left. The array at last will be the sorted array.

The only reason for using the merge sort algorithm for sorting the array is because it is almost 94% faster than the selection sort algorithm.

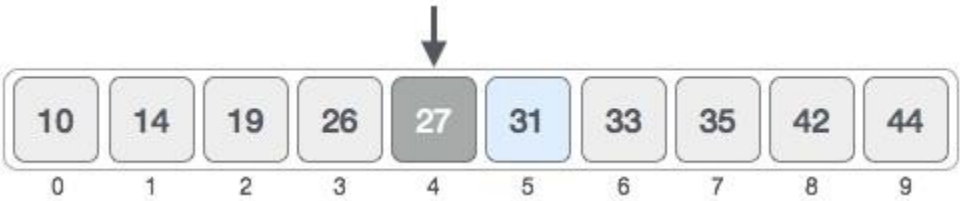
After sorting the array, sorted array is passed to the binary search algorithm for searching in the array.

Binary search is a fast search algorithm with run-time complexity of O(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form.

Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero.

For a binary search to work, the array where the value is to me search should be sorted. Let us consider a sorted array for using binary search algorithm to search 31 in the array

First of all, we shall determine the mid value of the array i.e. 4.5 which can be assumed that 4th index is the mid value.

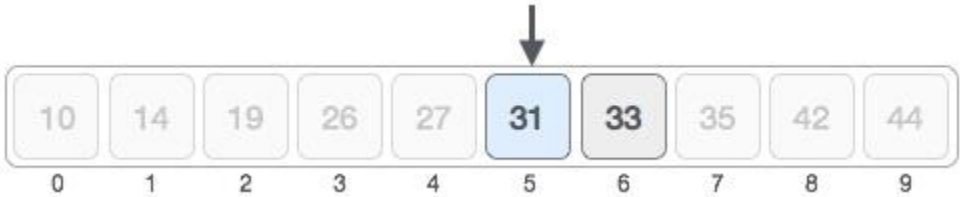


Now, comparing the value at 4th index with the search value '31'. As 31 is greater than 27, we know that the target portion of the array is portion on the right to 4th index.

Now, again we find the mid value of the target portion of the array which is 5+9=7. Here, 7th index is the mid value. So, we compare the search vale '31' with the mid value '35'.

The value stored at location 7 is not a match, rather it is more than what we are looking for. So, the value must be in the left side portion of the 7th index.

Hence, we calculate the mid again. This time it is 5th index.



We compare the value stored at location 5 with our target value. We find that it is a match. We conclude that the target value 31 is stored at location 5.

4. Method Description

4.1 MotoRacers()

This is the constructor of the class MotoRacers. In this constructor an 2D array of Object is created where the data of the bike in each array inside the main array. All the array inside the main array is written in the row representing one row.

4.2 btnAddActionPerformed (java.awt.event.ActionEvent evt)

This is the action performed method of the add button in the GUI. Here, all the user input from the input fields in the GUI are stored in the variables. They are validated and stored in an array. Then, the array is iterated and value is set in each column of a particular row.

4.3 btnExitActionPerformed()

This is the action performed method of the exit button in the GUI. This method is used to exit the application.

4.4 btnClearActionPerformed()

This is the action performed method of the clear button in the GUI. In this method, all the text fields, radio buttons and combo box are set to their default value.

4.5 btnUpdateActionPerformed()

This is the action performed method of the update button in the GUI. This method can be used to update the data of the bike displayed in the jTable. First of all, on clicking the update button an input is taken of the model number of which the data is to be updated. Then, the model number is checked if the bike of that particular bike number exists in the table, if exists a different dialog box appears to take the input for updating the attributes of the bike. After taking the input all the input are stored in an array and the array is set in the table.

4.6 delete1ActionPerformed()

This is the action performed method of the delete button in the GUI. This method can be used to delete a particular data of the bike displayed in the jTable. First of all, on clicking the delete button an input is taken of the model number of which the data is to be deleted. Then, the model number is checked if the bike of that particular bike number exists in the table. If the bike exist in the table then that bike detail is deleted.

4.7 sort(String[][] a)

This is a method in MergeSorter class. This method is used to break the array into sub arrays until there are only two elements left in the sub array. After that, the both elements are compared and set in ascending order.

4.8 merge(String[][] first, String[][] second, String[][] a)

This is a method in MergeSorter class. This method accepts two sorted sub arrays and another array where the two arrays first and second are merged are stored.

4.9 btnSearchActionPerformed()

This is the action performed method of the search button in the GUI. This method is used to search a particular bike data by using its price implementing the binary search algorithm using the merge sort algorithm. It divides the whole data into two parts and determines whether the searched value is in which part of the divided data by comparing with the middle value of the array and returns the lowest and highest index of the data.

4.10 search(String[][] a, int low, int high, int value)

This method accepts the 2D array, lowest index, highest index and the searched valued returned by the **btnSearchActionPerformed** method. This method searches the required data by breaking the array and comparing with the middle value until there are only two values left. If the desired result is obtained, it displays the data of the bike in an option pane.

4.11 itemOpenActionPerformed(java.awt.event.ActionEvent evt)

This method opens the documentation file of the project.

4.12 menuHelpActionPerformed(java.awt.event.ActionEvent evt)

This methods opens a pdf file which is a user guide to use the moto racers information systems application.

5. Testing

5.1 Running program

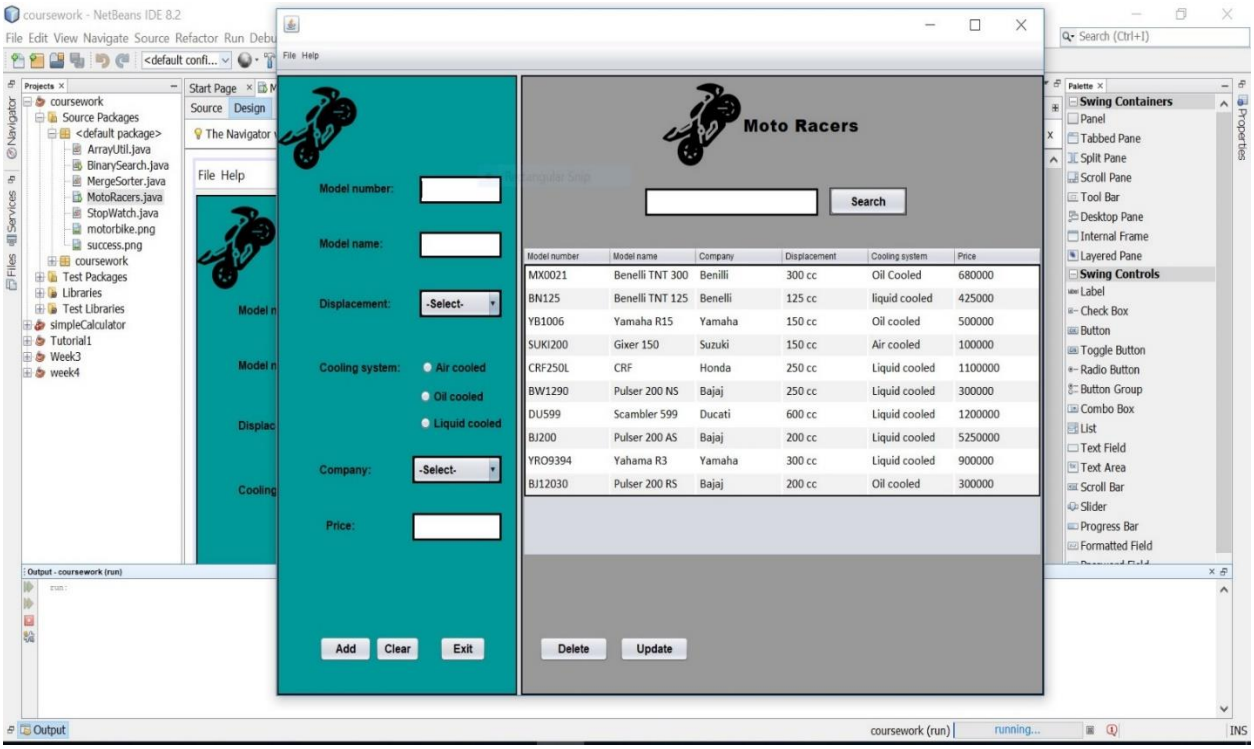


Figure 2: Test case 1: Running program successfully

Table 1: Test case 1

Test number	1
Action	Run the program
Expected output	When run is pressed the program should open.
Actual output	The program runs.
Test Result	Successful

5.2 Adding data to table

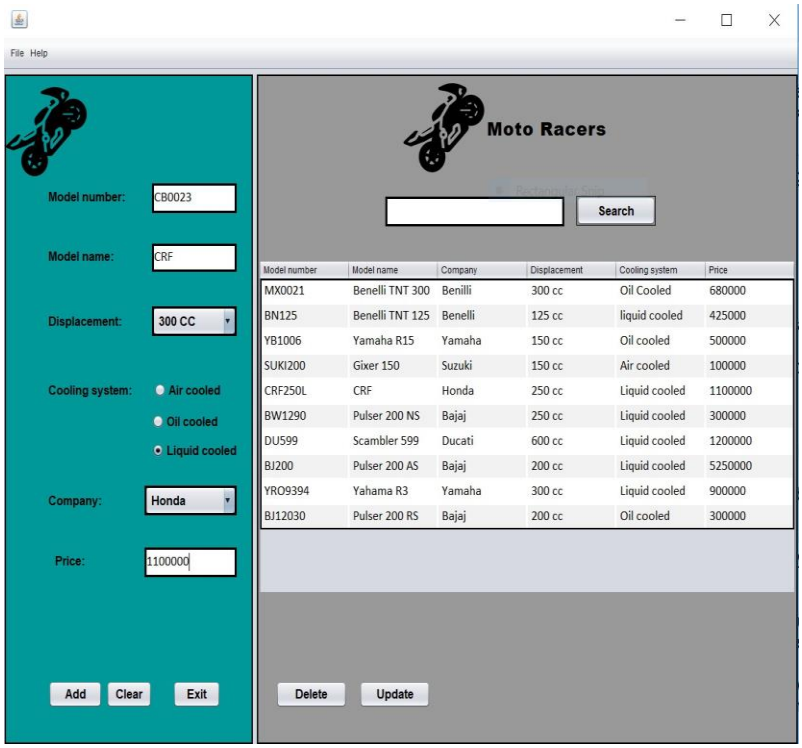


Figure 3: Before pressing add button

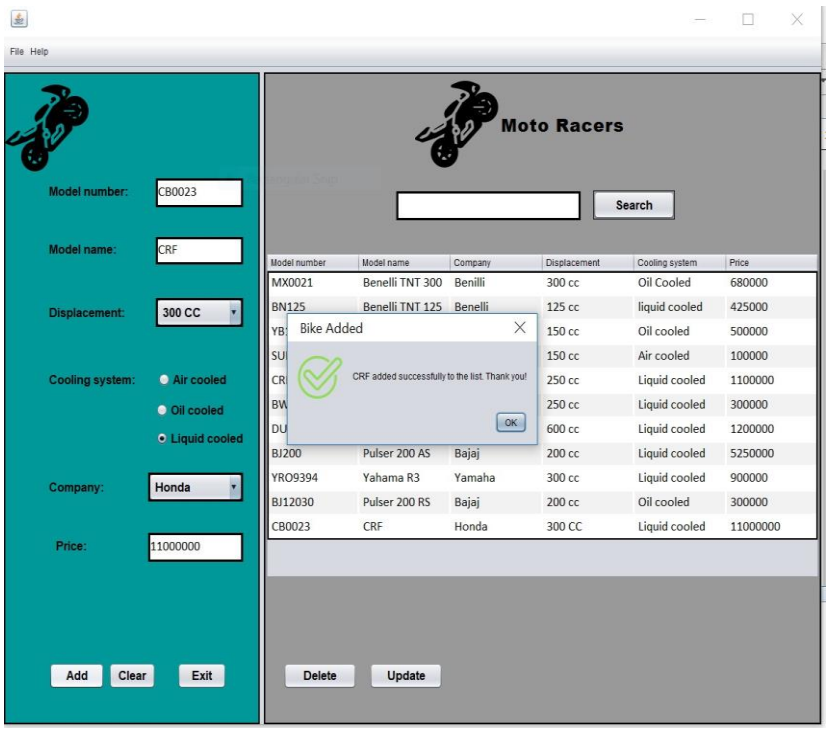


Figure 4: After pressing add button

Table 2: Test case 2

Test number	2
Action	Adding data to table
Expected output	The data is added to the table and dialog box stating the successful addition of the model number appears.
Actual output	The data is added to the table and dialog box with message appeared as expected.
Test Result	Successful

5.3 Searching for bike based on price

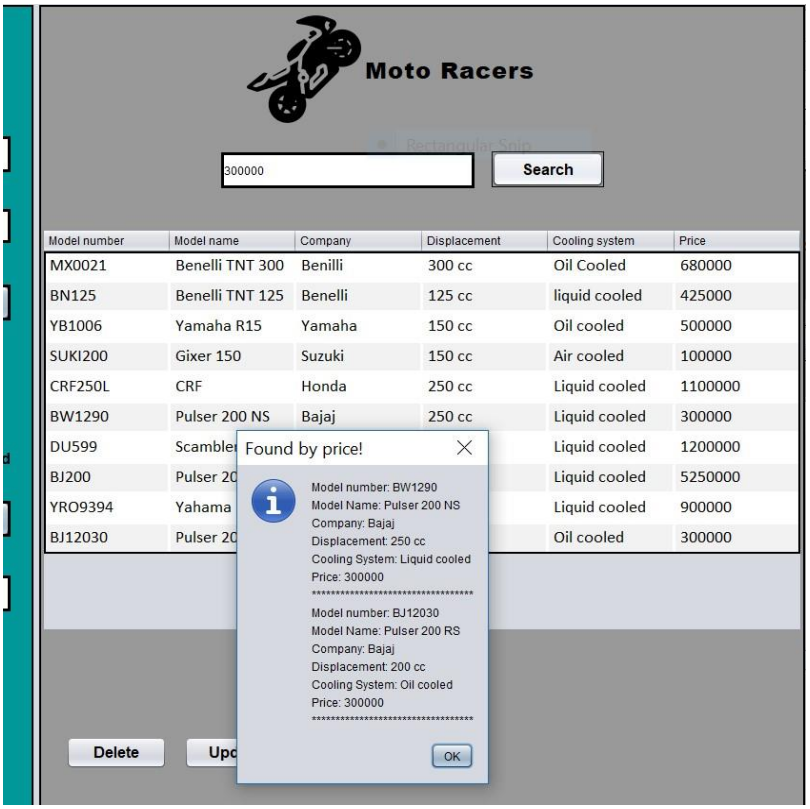


Figure 5: searching for bikes with price 300000

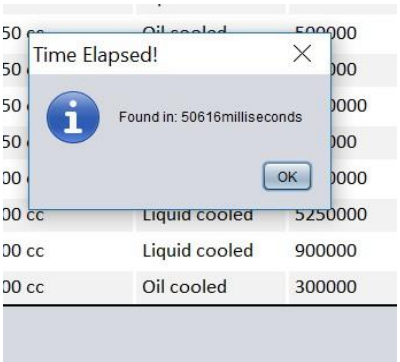


Figure 6: time elapsed for searching

Table 3: Test case 3

Test number	3
Action	Search bike according to the price
Expected output	The available bike with the given price is displayed. The time taken find the bikes is also displayed.
Actual output	The available bike with the given price is displayed. The time taken find the bikes is also displayed.
Test Result	Pass

5.4 Searching for number of bikes according to company

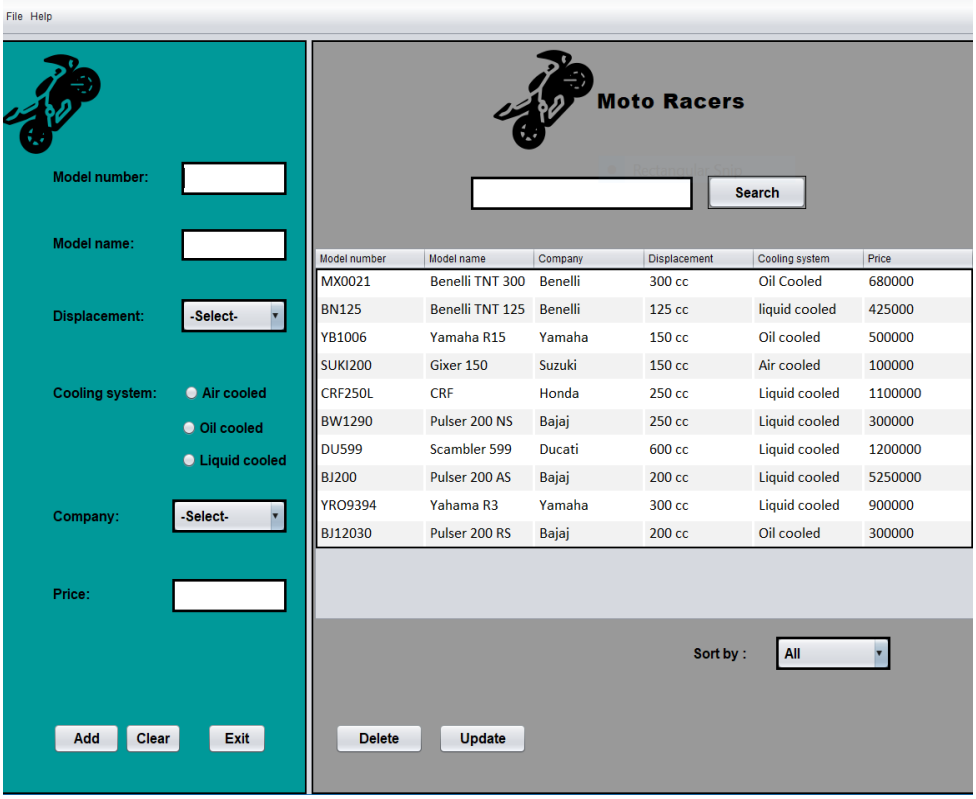


Figure 7: Before sorting the table

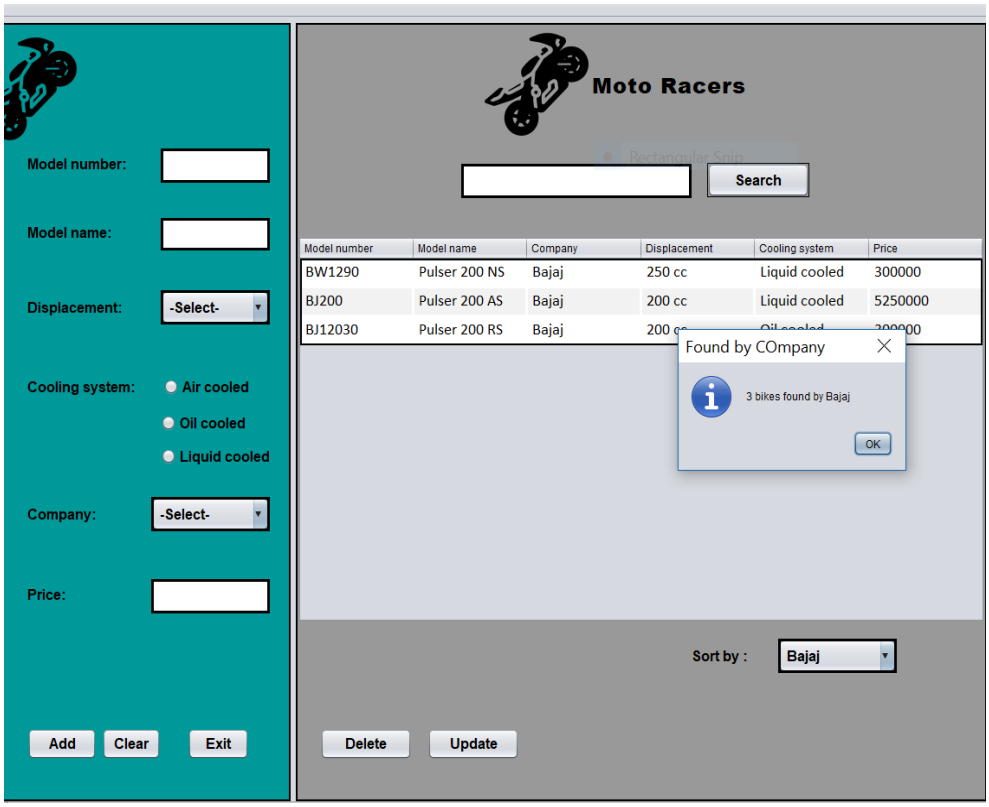


Figure 8: Sorting the table by Company 'Bajaj'

Table 4: Test case 4

Test number	4
Action	Searching for number of bikes according to the company name.
Expected output	The data of the bikes of selected company is displayed. The number of bikes available of that company is also displayed in a pop up message.
Actual output	The data of the bikes of selected company is displayed. The number of bikes available of that company is also displayed in a pop up message.
Test Result	Successful

5.5 Opening a file from open menu item

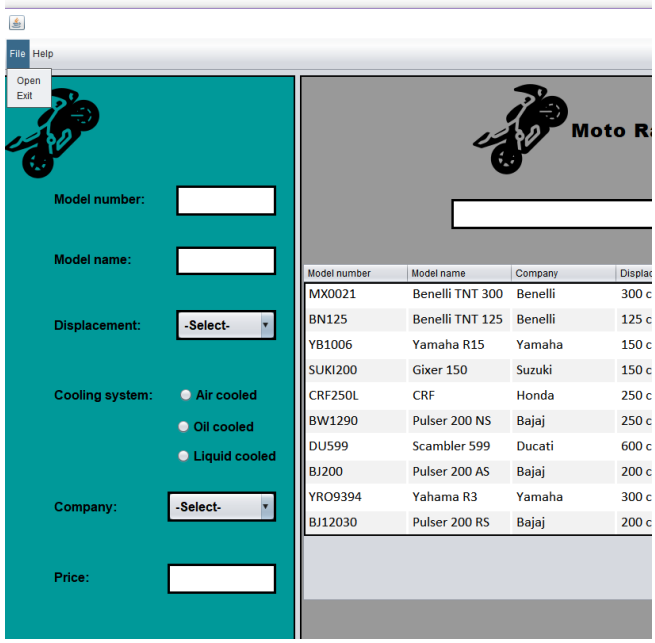


Figure 9: Before pressing open menu item:

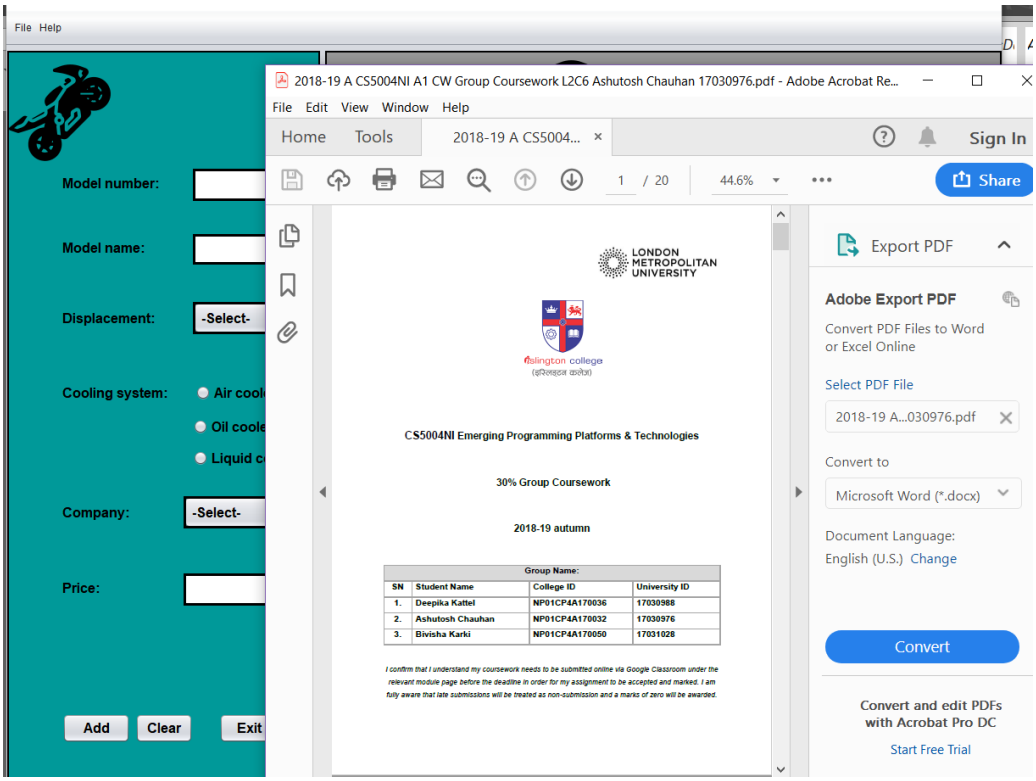


Figure 10: After pressing open menu item

Table 5: Test case 5

Test number	5
Action	Opening a file from menu.
Expected output	Project documentation pdf file will open
Actual output	Project documentation pdf file opened
Test Result	Successful

5.6 Negative value validation in search

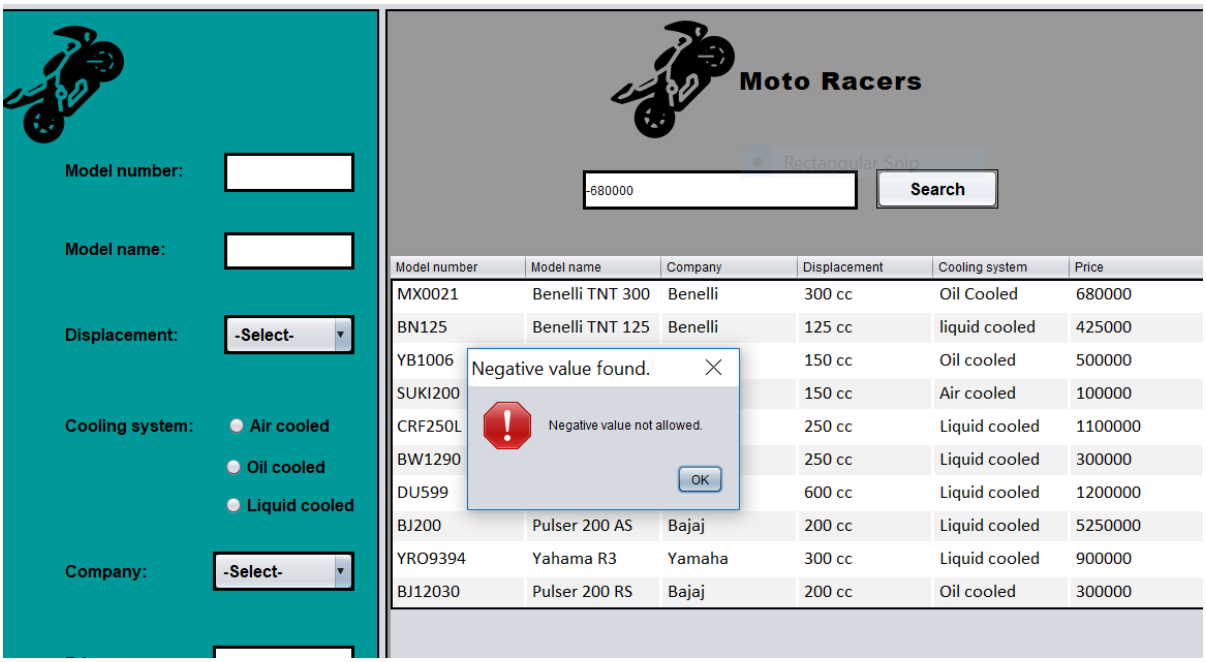


Figure 11: Negative value validation in search

Table 6: Test case 6

Test number	6
Action	Negative value is entered in the search box.
Expected output	Dialog box will be displayed showing that the entered value is negative and not allowed.
Actual output	Dialog box is displayed showing that the entered value is negative and not allowed.
Test Result	Successful

5.7 Number format validation in price

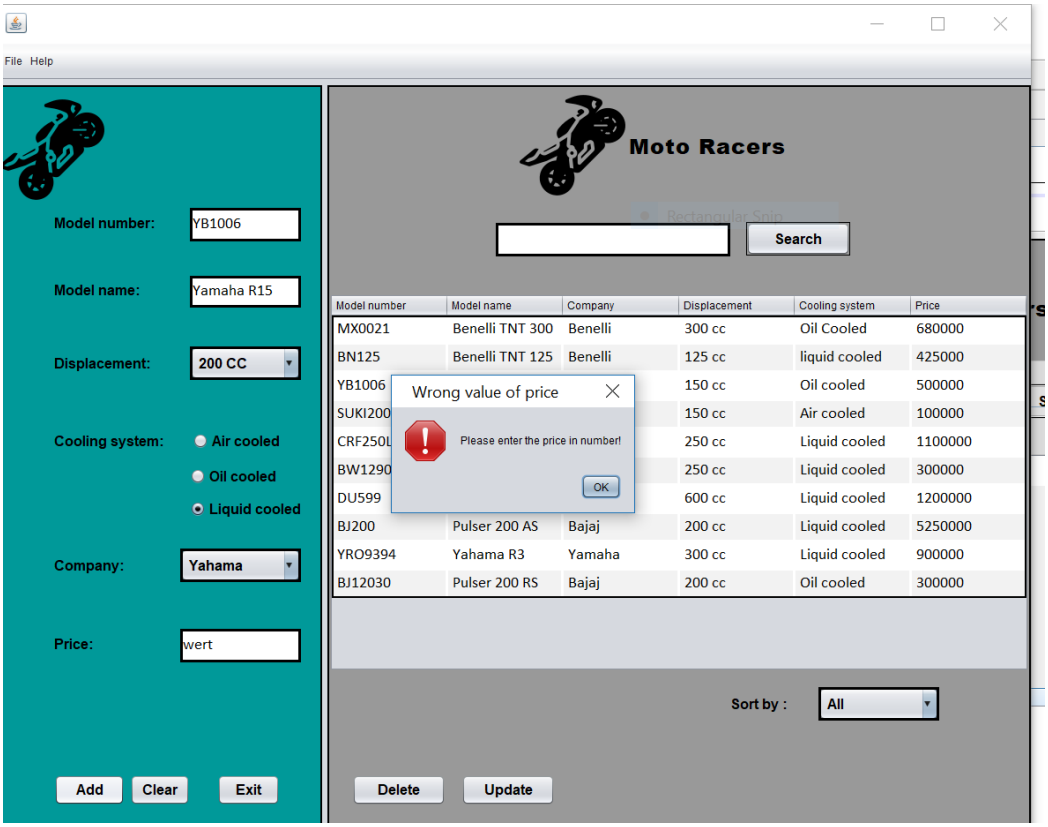


Figure 12: Number format validation in price

Table 7: Test case 7

Test number	7
Action	String value is entered in the price box.
Expected output	Dialog box will be displayed showing that the entered price is string and not allowed.
Actual output	Dialog box is displayed showing that the entered price is string and not allowed.
Test Result	Successful

5.8 Duplicate data entry in the table

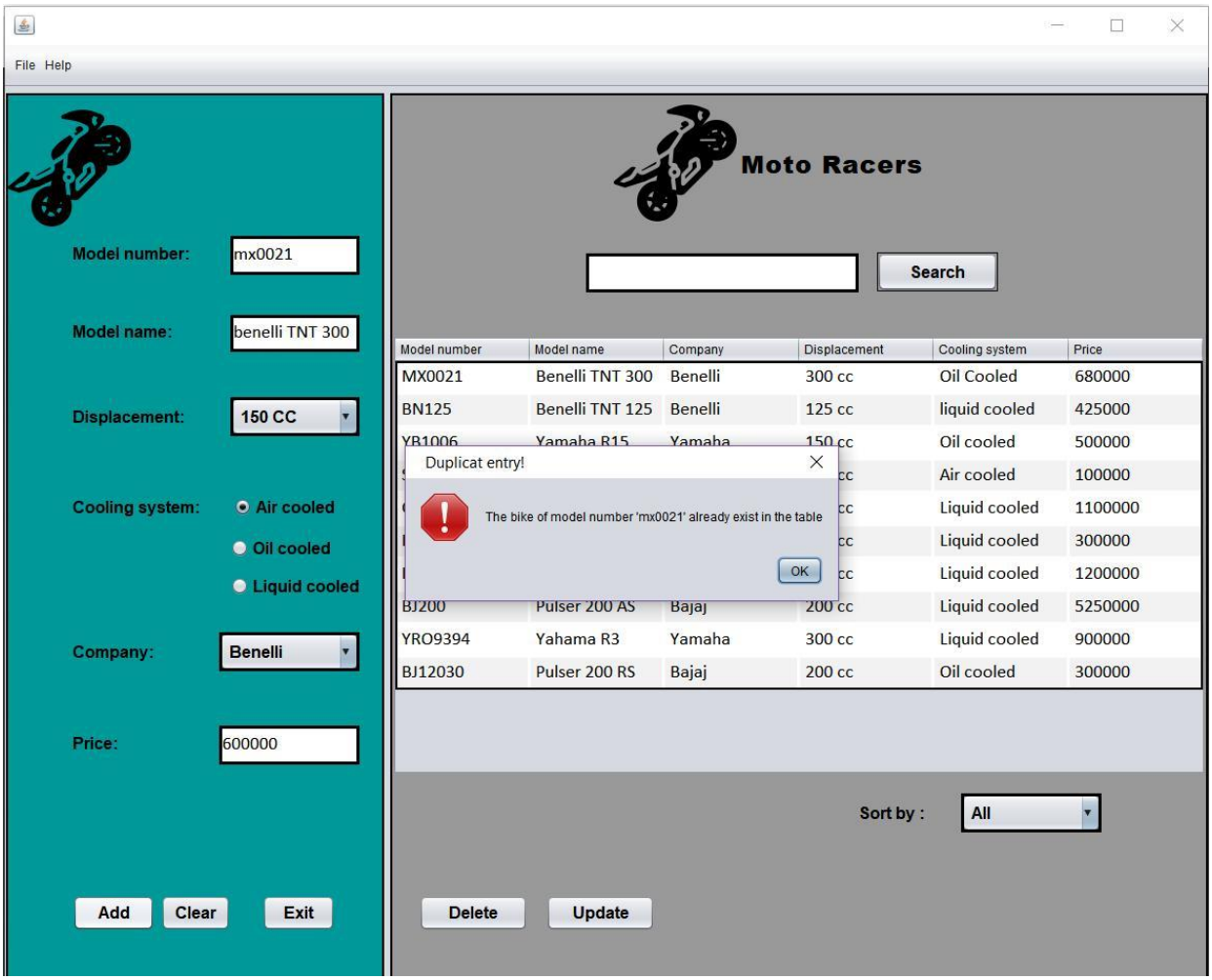


Figure 13: Duplicate data entry in the table

Table 8: Test case 8

Test number	8
Action	Duplicate data is entered.
Expected output	Dialog box will be displayed showing that the entered data already exists.
Actual output	Dialog box is displayed showing that the entered data already exists.

Test Result	Successful

5.9 Empty text field validation

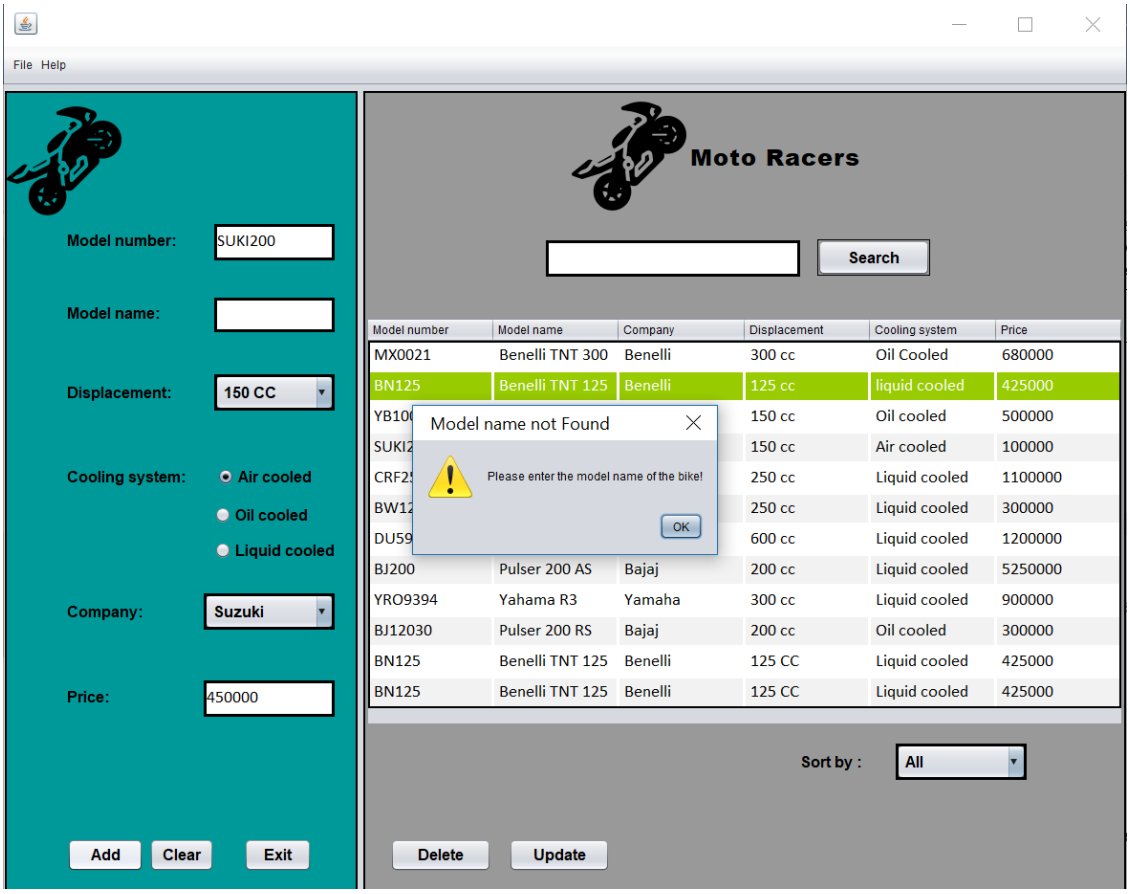


Figure 14: empty text field validation

Table 9: Test case 9

Test number	9
Action	Text field is left empty.
Expected output	Dialog box will be displayed showing that text field is left empty.
Actual output	Dialog box is displayed showing that text field is left empty.
Test Result	Successful

5.10 Updating existing data

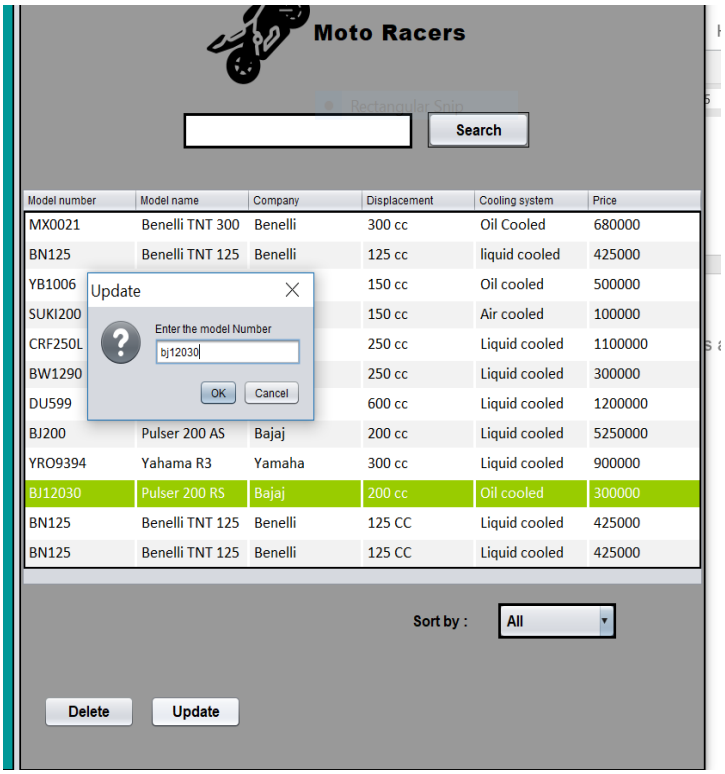


Figure 16: Entering model number to update it's existing data

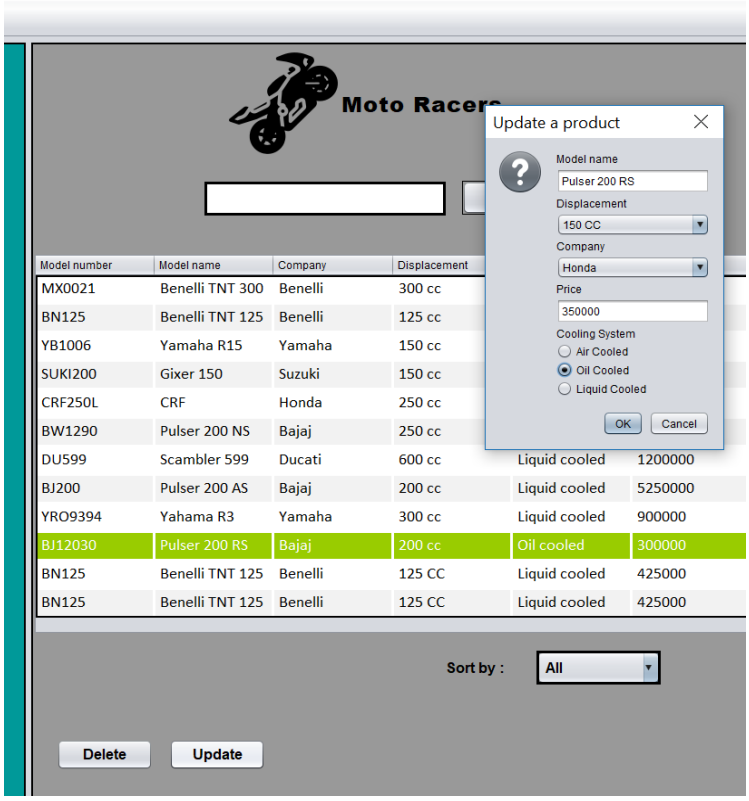


Figure 15: Entering the updated data of the entered model number

Figure 17: Displaying updated result in the table

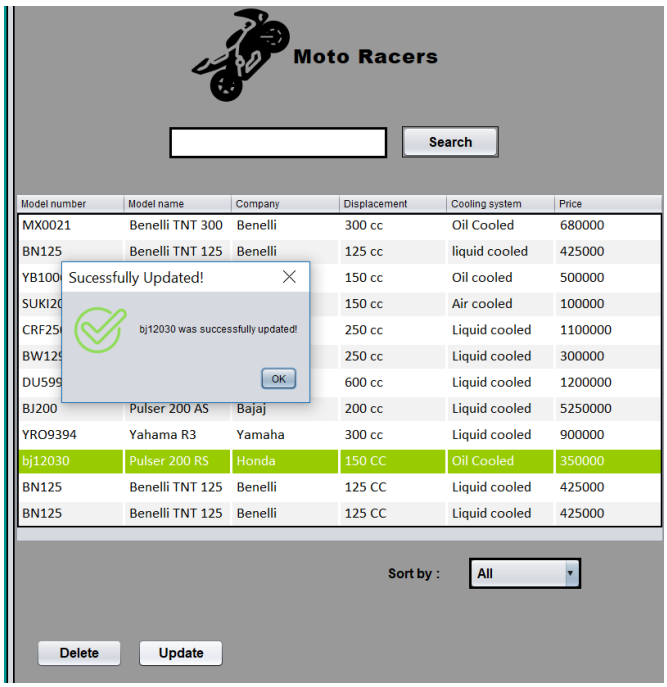


Table 10: Test case 10

Test number	10
Action	1. Update button is pressed. 2. Model number of bike to be updated is entered. 3. Data to be updated is entered.
Expected output	Data will be updated.
Actual output	Data is updated.
Test Result	Successful

5.11 Deleting a data from the table

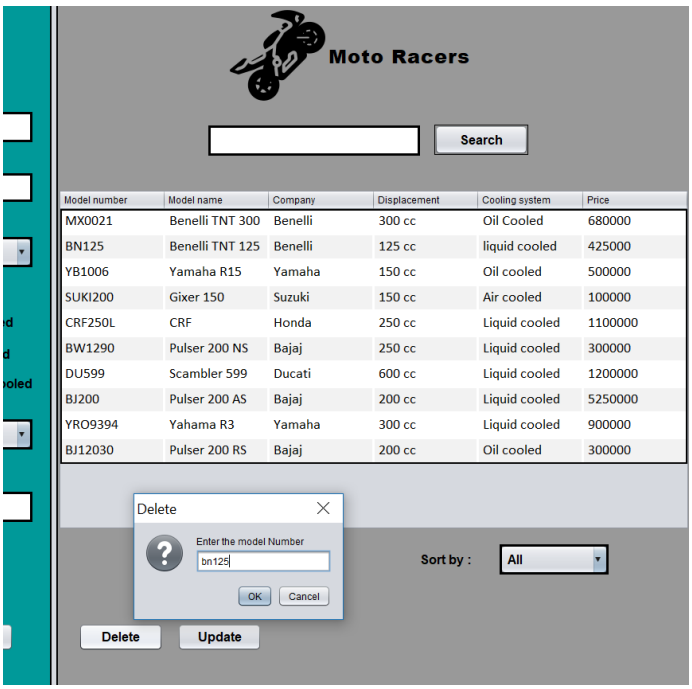


Figure 19: Entering the model number of bike to be deleted

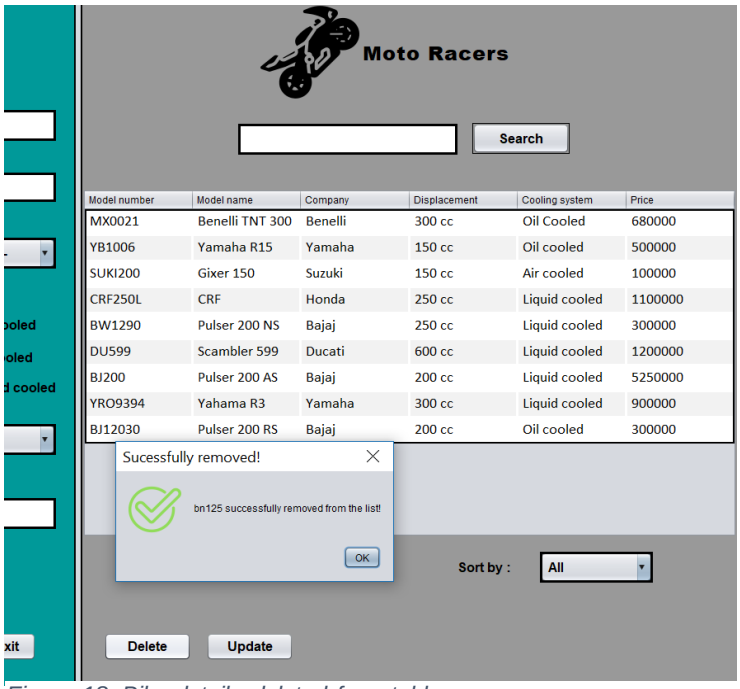


Figure 18: Bike details deleted from table

Table 11: Test case 11

Test number	11
Action	1. Delete button is pressed. 2. Model number of bike to be deleted is entered.
Expected output	Data will be deleted.
Actual output	Data is deleted.
Test Result	Successful

5.12 Logical error: Adding bike to the list with negative price

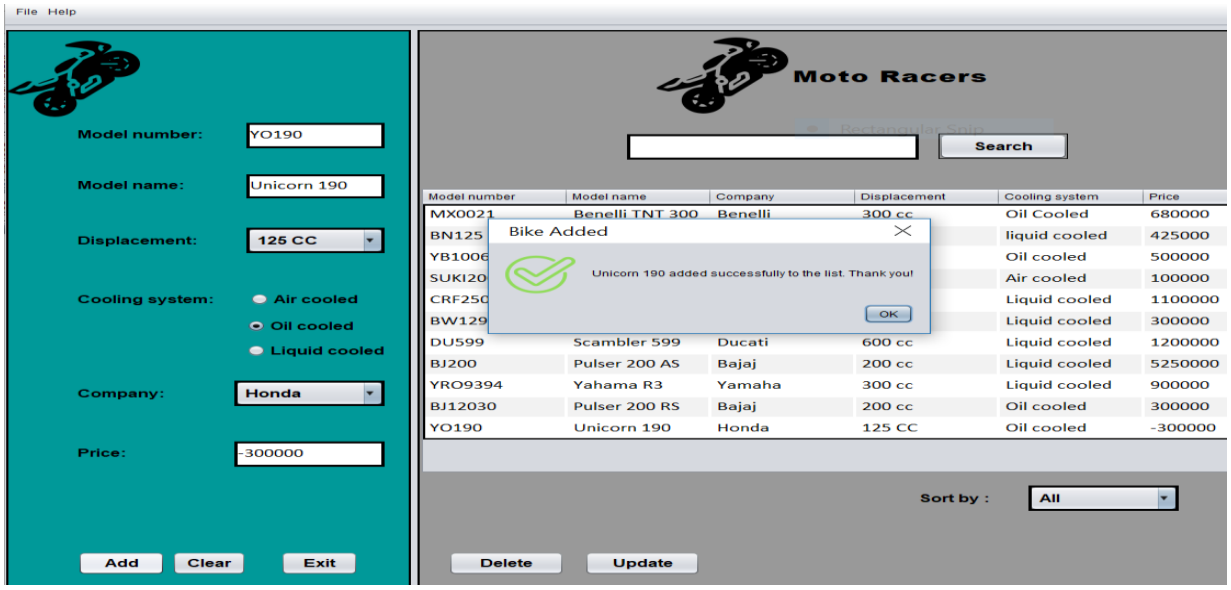


Figure 20: Adding bike with negative price

Table 12: Test case 12

Test number	12
Action	Adding bike details with negative price.
Expected output	Dialog box should be displayed with suitable message.
Actual output	Bike details added to the table.
Test Result	Logical error detected.

5.13 Logical error: deleting the bike if the model number entered in lower case

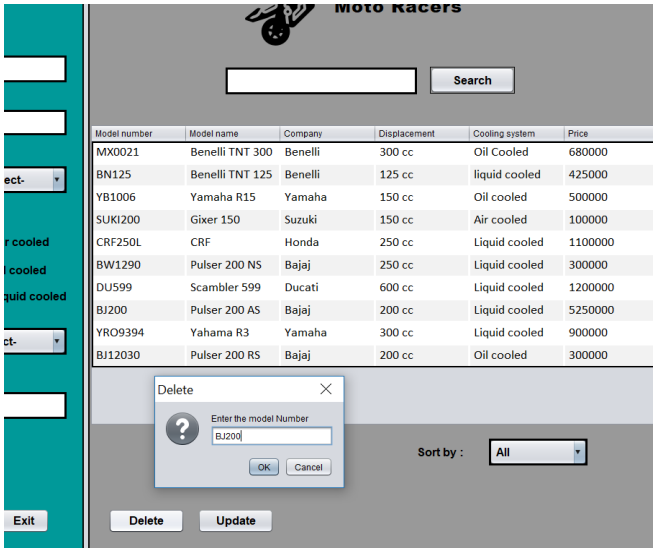


Figure 22: Entering the bike model number in upper case

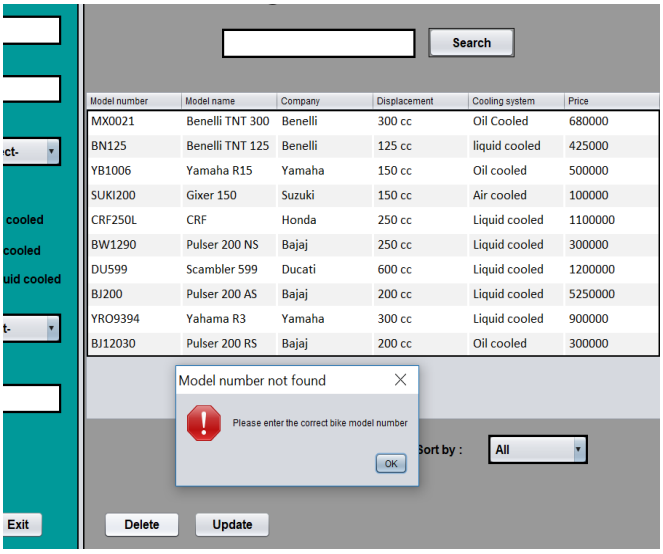


Figure 21: Error message displayed

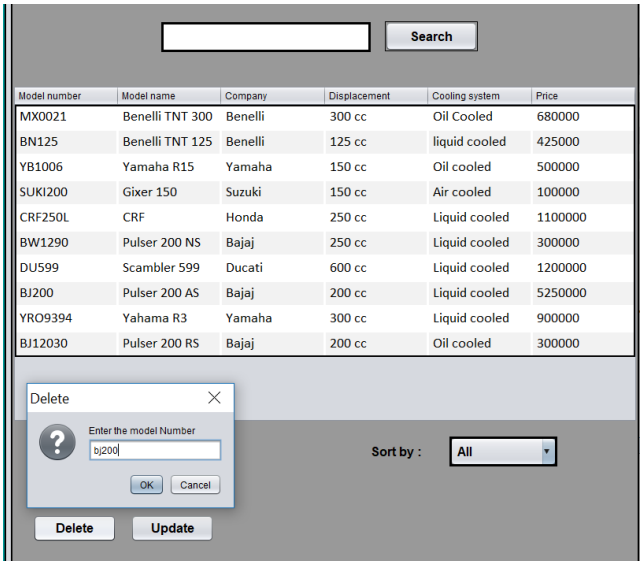


Figure 24: Entering the same bike model number in lower case

Figure 23: Bike details deleted

Table 13: Test case 13

Test number	13
Action	Model number of bike is entered in upper and then in lower case to be deleted.
Expected output	Model number in both upper case and lower case should be deleted.
Actual output	Model number in lower case is only deleted.
Test Result	Logical error detected.

6. Conclusion

The project that we are required to develop here in this module covers almost every area of study that we practiced and learned this entire semester. The project helped us understand more about the topics as we could share ideas among our group members. The implementation of those ideas in our project has helped us clear our vision about the areas that we have to implement here in this project.

The project is developed of the processes that we require for the development of a software in real life. The scenario is to develop a software for a fitness gym. The software records all the details of the customers, staff, the daily activities and updates everything time to time. The data flow diagrams, structure charts, process specification, module specification were taught to us and here in this project we are to include every aspect that we learned and develop this project accordingly. The linking of the scenario with the required process that we have to implement here helped us widen out knowledge about these topics.

The development of this project was not easy. However, with the effort of every group member we could overcome the difficulties. The support of the entire group helped us complete this project on time. This project in general helped us work in team although every single of us have conflicting ideas.

7. References

java2novice, 2019. *Program: Implement merge sort in java..* [Online] Available at: http://www.java2novice.com/java-sorting-algorithms/merge-sort/ [Accessed 25 01 2019].

tutorialspoint, 2019. *Data Structure and Algorithms Binary Search.* [Online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm [Accessed 21 January 2019].

8. Appendix

```
import java.awt.Color;
import java.awt.Container;
import java.awt.Desktop;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import static java.lang.Integer.parseInt;
import static java.lang.Integer.parseInt;
import java.lang.reflect.Array;
import java.net.URL;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.PatternSyntaxException;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
```

```
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JTextField;
import javax.swing.RowFilter;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

/*
 * 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 Deepika Kattel
 */
public class MotoRacers extends javax.swing.JFrame {
    //declaring success image and input image
   ImageIcon icon_success = new ImageIcon("checked.png");
   ImageIcon icon_input = new ImageIcon("input.png");
   /**
```

```
 * Creates new form MotoRacers
 */
public MotoRacers() {



    initComponents();
    //creating 2d array of object for setting the dummy data in the jTable
    Object[][] defaultValue = {
        {"MX0021", "Benelli TNT 300", "Benelli", "300 cc", "Oil Cooled", "680000"},
        {"BN125", "Benelli TNT 125", "Benelli", "125 cc", "liquid cooled", "425000"},
        {"YB1006", "Yamaha R15", "Yamaha", "150 cc", "Oil cooled", "500000"},
        {"SUKI200", "Gixer 150", "Suzuki", "150 cc", "Air cooled", "100000"},
        {"CRF250L", "CRF", "Honda", "250 cc", "Liquid cooled", "1100000"},
        {"BW1290", "Pulser 200 NS", "Bajaj", "250 cc", "Liquid cooled", "300000"},
        {"DU599", "Scambler 599", "Ducati", "600 cc", "Liquid cooled", "1200000"},
        {"BJ200", "Pulser 200 AS", "Bajaj", "200 cc", "Liquid cooled", "5250000"},
        {"YRO9394", "Yahama R3", "Yamaha", "300 cc", "Liquid cooled", "900000"},
        {"BJ12030", "Pulser 200 RS", "Bajaj", "200 cc", "Oil cooled", "300000"}
    };

    DefaultTableModel model1 = (DefaultTableModel) jTable1.getModel();
    //setting the default values in the jTable
    for (int i = 0; i < defaultValue.length; i++) {
        model1.addRow(defaultValue[i]);
```

```
    }
}

/**
 * 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() {

    cooling = new javax.swing.ButtonGroup();
    fileChooser = new javax.swing.JFileChooser();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    model_num = new javax.swing.JTextField();
    displacement = new javax.swing.JComboBox<>();
    jLabel3 = new javax.swing.JLabel();
    coolAir = new javax.swing.JRadioButton();
    coolOil = new javax.swing.JRadioButton();
    coolLiquid = new javax.swing.JRadioButton();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
```

```
btnAdd = new javax.swing.JButton();
btnClear = new javax.swing.JButton();
btnExit = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
model_name = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
price = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
company1 = new javax.swing.JComboBox<>();
jPanel3 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
btnUpdate = new javax.swing.JButton();
searchBar = new javax.swing.JTextField();
btnSearch = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
delete1 = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();
companyCmb = new javax.swing.JComboBox<>();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
itemOpen = new javax.swing.JMenuItem();
itemexit = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
```

```
        menuHelp = new javax.swing.JMenuItem();

        fileChooser.setPreferredSize(new java.awt.Dimension(700, 600));

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setBackground(new java.awt.Color(255, 102, 102));

        jPanel1.setBackground(new java.awt.Color(0, 153, 153));
        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 2));

        jLabel2.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
        jLabel2.setText("Model number:");

        model_num.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N
        model_num.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));
        model_num.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                model_numActionPerformed(evt);
            }
        });

        displacement.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
        displacement.setForeground(new java.awt.Color(1, 1, 1));
```

```
        displacement.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "-Select-", "100 CC", "125 CC", "150 CC", "180 CC", "200 CC", "300 CC", "400 CC", "500 CC" }));

        displacement.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0), 3));

        displacement.addItemListener(new java.awt.event.ItemListener() {

            public void itemStateChanged(java.awt.event.ItemEvent evt) {

                displacementItemStateChanged(evt);

            }

        });


        jLabel3.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

        jLabel3.setText("Displacement:");


        coolAir.setBackground(new java.awt.Color(0, 153, 153));

        cooling.add(coolAir);

        coolAir.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

        coolAir.setText("Air cooled");

        coolAir.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0), 3));


        coolOil.setBackground(new java.awt.Color(0, 153, 153));

        cooling.add(coolOil);

        coolOil.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

        coolOil.setText("Oil cooled");

        coolOil.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            coolOilActionPerformed(evt);
        }
    });

    coolLiquid.setBackground(new java.awt.Color(0, 153, 153));
    cooling.add(coolLiquid);
    coolLiquid.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
    coolLiquid.setText("Liquid cooled");
    coolLiquid.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            coolLiquidActionPerformed(evt);
        }
    });

    jLabel4.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
    jLabel4.setText("Cooling system:");

    jLabel5.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
    jLabel5.setText("Company:");

    btnAdd.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
    btnAdd.setText("Add");
    btnAdd.setBorder(null);
```

```
btnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddActionPerformed(evt);
    }
});

btnClear.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
btnClear.setText("Clear");
btnClear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnClearActionPerformed(evt);
    }
});

btnExit.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
btnExit.setText("Exit");
btnExit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnExitActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
jLabel1.setText("Model name:");
```

```
model_name.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

model_name.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));


jLabel6.setFont(new java.awt.Font("Arial Black", 1, 24)); // NOI18N

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/motorbike.png"))); // NOI18N


price.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N

price.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
0, 0), 3));

price.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        priceActionPerformed(evt);

    }

});


jLabel7.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

jLabel7.setText("Price:");


company1.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

company1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "-
Select-", "Suzuki", "Honda", "Bajaj", "KTM", "Benelli", "BMW", "Ducati", "Yahama" }));

company1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));
```

```
javax.swing.GroupLayout  jPanel1Layout  = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(63, 63, 63)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                        .addComponent(jLabel2)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                            .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    148,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel3)
                            .addComponent(jLabel4)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
```

```
                    .addComponent(coolLiquid,
javax.swing.GroupLayout.DEFAULT_SIZE,   javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                    .addComponent(model_name)

                    .addComponent(model_num)

                    .addComponent(displacement,                                  0,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE)

                    .addComponent(coolAir,    javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE)

                    .addComponent(coolOil,    javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE))

                .addGap(0, 0, Short.MAX_VALUE))

            .addGroup(jPanel1Layout.createSequentialGroup()


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

                    .addComponent(jLabel5)

                    .addComponent(btnAdd,
javax.swing.GroupLayout.PREFERRED_SIZE,                                          80,
javax.swing.GroupLayout.PREFERRED_SIZE))


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)

                    .addGroup(jPanel1Layout.createSequentialGroup()


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                        .addComponent(btnClear)

                        .addGap(31, 31, 31)
```

```
                    .addComponent(btnExit,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    71,
javax.swing.GroupLayout.PREFERRED_SIZE))

                   .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE)

                    .addComponent(company1,
javax.swing.GroupLayout.PREFERRED_SIZE,                                   137,
javax.swing.GroupLayout.PREFERRED_SIZE))))

              .addGroup(jPanel1Layout.createSequentialGroup()

                 .addComponent(jLabel7)


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE)

                  .addComponent(price,      javax.swing.GroupLayout.PREFERRED_SIZE,
137, javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addContainerGap(22, Short.MAX_VALUE))

         .addGroup(jPanel1Layout.createSequentialGroup()

            .addComponent(jLabel6,  javax.swing.GroupLayout.PREFERRED_SIZE,  137,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(0, 0, Short.MAX_VALUE))

      );

      jPanel1Layout.setVerticalGroup(


jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

            .addGap(11, 11, 11)
```

```
            .addComponent(jLabel6)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)

                .addComponent(jLabel2)

                .addComponent(model_num,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    38,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(39, 39, 39)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)

                .addComponent(jLabel1)

                .addComponent(model_name,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    36,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(44, 44, 44)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)

                .addComponent(jLabel3)

                .addComponent(displacement,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    38,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(57, 57, 57)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)

                .addComponent(coolAir)
```

```
                .addComponent(jLabel4))
            .addGap(18, 18, 18)
            .addComponent(coolOil)
            .addGap(18, 18, 18)
            .addComponent(coolLiquid)
            .addGap(35, 35, 35)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(jLabel5)
                .addComponent(company1,  javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(53, 53, 53)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(price,   javax.swing.GroupLayout.PREFERRED_SIZE,   38,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel7))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(btnClear,    javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnExit, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
            .addGap(44, 44, 44))
        );

        jPanel3.setBackground(new java.awt.Color(153, 153, 153));
        jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 2));

        jTable1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 2));
        jTable1.setFont(new java.awt.Font("Calibri", 0, 18)); // NOI18N
        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Model number", "Model name", "Company", "Displacement", "Cooling
system", "Price"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, true, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
```

```
        });
        jTable1.setAlignmentX(1.0F);
        jTable1.setAlignmentY(1.0F);
        jTable1.setIntercellSpacing(new java.awt.Dimension(3, 3));
        jTable1.setRowHeight(32);
        jTable1.setSelectionBackground(new java.awt.Color(153, 204, 0));
        jScrollPane1.setViewportView(jTable1);

        btnUpdate.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
        btnUpdate.setText("Update");
        btnUpdate.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnUpdateActionPerformed(evt);
            }
        });

        searchBar.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));
        searchBar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                searchBarActionPerformed(evt);
            }
        });

        btnSearch.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N
```

```
        btnSearch.setText("Search");

        btnSearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        btnSearch.addActionListener(new  java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent  evt) {

                btnSearchActionPerformed(evt);

            }

        });


        jLabel8.setFont(new java.awt.Font("Arial Black", 1, 24)); // NOI18N

        jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/motorbike.png")));  // NOI18N

        jLabel8.setText("Moto Racers");


        delete1.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

        delete1.setText("Delete");

        delete1.addActionListener(new  java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent  evt) {

                delete1ActionPerformed(evt);

            }

        });


        jLabel9.setFont(new java.awt.Font("Arial", 1, 16)); // NOI18N

        jLabel9.setText("Sort by :");
```

```
        companyCmb.setFont(new  java.awt.Font("Arial",  1, 16)); // NOI18N

        companyCmb.setModel(new  javax.swing.DefaultComboBoxModel<>(new  String[] {
"All", "Suzuki",  "Honda",  "Bajaj", "KTM", "Benelli",  "BMW", "Ducati", "Yahama" }));

        companyCmb.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0), 3));

        companyCmb.addActionListener(new  java.awt.event.ActionListener()  {

            public void actionPerformed(java.awt.event.ActionEvent  evt) {

                companyCmbActionPerformed(evt);

            }

        });


        javax.swing.GroupLayout  jPanel3Layout  = new javax.swing.GroupLayout(jPanel3);

        jPanel3.setLayout(jPanel3Layout);

        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jScrollPane1)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

                .addGap(0, 0, Short.MAX_VALUE)

                .addComponent(jLabel9)

                .addGap(35, 35, 35)

                .addComponent(companyCmb,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    137,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(102, 102, 102))

            .addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGap(26, 26, 26)
                    .addComponent(delete1,   javax.swing.GroupLayout.PREFERRED_SIZE,
106, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    106,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGap(189, 189, 189)
                    .addComponent(searchBar,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    266,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnSearch,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    118,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGap(216, 216, 216)
                    .addComponent(jLabel8,   javax.swing.GroupLayout.PREFERRED_SIZE,
328, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(203, Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel8)
            .addGap(27, 27, 27)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(searchBar,  javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnSearch,  javax.swing.GroupLayout.PREFERRED_SIZE,
38, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(42, 42, 42)
            .addComponent(jScrollPane1,  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(jLabel9)
                .addComponent(companyCmb,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    38,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
61, Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(delete1,     javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addComponent(btnUpdate, javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(44, 44, 44))
        );

        jMenuBar1.setMaximumSize(new java.awt.Dimension(200, 32769));
        jMenuBar1.setPreferredSize(new java.awt.Dimension(80, 40));

        jMenu1.setText("File");
        jMenu1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu1ActionPerformed(evt);
            }
        });

        itemOpen.setText("Open");
        itemOpen.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                itemOpenActionPerformed(evt);
            }
        });
        jMenu1.add(itemOpen);

        itemexit.setText("Exit");
        itemexit.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            itemexitActionPerformed(evt);
        }
    });
    jMenu1.add(itemexit);

    jMenuBar1.add(jMenu1);

    jMenu2.setText("Help");
    jMenu2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenu2ActionPerformed(evt);
        }
    });

    menuHelp.setText("Help file");
    menuHelp.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            menuHelpActionPerformed(evt);
        }
    });
    jMenu2.add(menuHelp);

    jMenuBar1.add(jMenu2);
```

```
setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(1, 1, 1)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(7, 7, 7)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
```

```
        );

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

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

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

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

    //Method to add data in the table
    private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
        //Fetching the data entered in the text fields
        String model_number1 = model_num.getText();
        String model_name1 = model_name.getText();
        String price1;
        int price2 = 0;
```

```
String price3 = "";
//Validating ther user input data and adding to the table
if (!model_number1.equals("")) {
    if (!model_name1.equals("")) {
        if (displacement.getSelectedIndex() != 0) {
            if (!cooling.isSelected(null)) {
                if (company1.getSelectedIndex() != 0) {
                    price3 = price.getText();
                    if (!price3.equals("")) {

                        try {
                            price2 = Integer.parseInt(price3);
                            if (price2 > 0) {
                                int rowCount = jTable1.getRowCount();
                                String a = "";
                                //creating a linked list to store the if there is repeated value of
the entered
                                LinkedList repeat = new LinkedList();
                                for (int i = 0; i < rowCount; i++) {
                                    String model_number2 = jTable1.getValueAt(i,
0).toString().toLowerCase();
                                    if (model_number2.equals(model_number1)) {
                                        //adding repeated data in repeat linkedlist
                                        repeat.add(model_number2);
                                    }
```

```
                    }
                    //if there is no repeated data in the repeat list, data is added to
the table
                    if (repeat.size() == 0) {
                        price1 = String.valueOf(price2);
                        String cooling1;
                        String displacement1 =
displacement.getSelectedItem().toString();
                        String company2 = company1.getSelectedItem().toString();
                        if (coolAir.isSelected()) {
                            cooling1 = coolAir.getText();
                        } else if (coolLiquid.isSelected()) {
                            cooling1 = coolLiquid.getText();
                        } else {
                            cooling1 = coolOil.getText();
                        }
                        String[] bikeDetails = {model_number1, model_name1,
company2, displacement1, cooling1, price1};
                        DefaultTableModel model1 = (DefaultTableModel)
jTable1.getModel();
                        Object[] add={null};
                        model1.addRow(add);
                        try{
                            for (int i=0;i<bikeDetails.length;i++){
                                jTable1.setValueAt(bikeDetails[i], rowCount, i);
                            }
```

```
                    }

                    catch(ArrayIndexOutOfBoundsException e){};

                    JOptionPane.showMessageDialog(rootPane, model_name1 + " added successfully to the list. Thank you!", " Bike Added", JOptionPane.INFORMATION_MESSAGE, icon_success);

                } else {

                    JOptionPane.showMessageDialog(rootPane, "The bike of model number '" + model_number1 + "' already exist in the table", " Duplicat entry!", JOptionPane.ERROR_MESSAGE);

                }

            } else {

                JOptionPane.showMessageDialog(rootPane, "The price of the bike should be greater than 0!", " Wrong value of price", JOptionPane.ERROR_MESSAGE);

            }


        } catch (NumberFormatException e) {

            JOptionPane.showMessageDialog(rootPane, "Please enter the price in number!", " Wrong value of price", JOptionPane.ERROR_MESSAGE);

        }

    } else {

        JOptionPane.showMessageDialog(rootPane, "Please enter the price of the bike!", " Price not found", JOptionPane.ERROR_MESSAGE);

    }


} else {

    JOptionPane.showMessageDialog(rootPane, "Please select the company of bike!", " Company not found", JOptionPane.WARNING_MESSAGE);
```

```
                }
            } else {
                JOptionPane.showMessageDialog(rootPane, "Please select the cooling system of the bike!", " Cooling System not found", JOptionPane.WARNING_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(rootPane, "Please select the displacement of the bike!", " Diplacement not dound", JOptionPane.WARNING_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(rootPane, "Please enter the model name of the bike!", "  Model name not Found", JOptionPane.WARNING_MESSAGE);
    }


  } else {
    JOptionPane.showMessageDialog(rootPane, "Please enter the model bumber of the bike!", "   Model number not Found", JOptionPane.WARNING_MESSAGE);
  }
}


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

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


    }


    private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
        //clearing all the text fields, combo box and radio button to default value
        model_num.setText("");
        model_name.setText("");
        price.setText("");
        displacement.setSelectedIndex(0);
        cooling.clearSelection();
        company1.setSelectedIndex(0);
    }



    private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        String model_num2 = "";


        boolean loop = false;
        while (loop == false) {
            try {
                JTextField model_num3 = new JTextField();
                Object update[] = {"Enter the model Number", model_num3};
                int a = JOptionPane.showConfirmDialog(rootPane, update, "Update",
JOptionPane.OK_CANCEL_OPTION);
```

```
model_num2 = model_num3.getText().toLowerCase();

if (a == JOptionPane.OK_OPTION) {

    if (!model_num2.equals("")) {
        JTextField model_name2 = new JTextField();
        JTextField price2 = new JTextField();

        JComboBox<String> displacement2 = new JComboBox<String>();
        displacement2.addItem("100 CC");
        displacement2.addItem("125 CC");
        displacement2.addItem("150 CC");
        displacement2.addItem("180 CC");
        displacement2.addItem("200 CC");
        displacement2.addItem("300 CC");
        displacement2.addItem("400 CC");
        displacement2.addItem("500 CC");

        JComboBox<String> company2 = new JComboBox<String>();
        company2.addItem("Suzuki");
        company2.addItem("Honda");
        company2.addItem("Bajaj");
        company2.addItem("KTM");
        company2.addItem("Benelli");
```

```
company2.addItem("BMW");
company2.addItem("Ducati");
company2.addItem("Yamaha");

ButtonGroup group = new ButtonGroup();
JRadioButton coolAir2 = new JRadioButton("Air Cooled");
JRadioButton coolOil2 = new JRadioButton("Oil Cooled");
JRadioButton coolLiquid2 = new JRadioButton("Liquid Cooled");
group.add(coolAir2);
group.add(coolOil2);
group.add(coolLiquid2);

Object[] update2 = {
    "Model name", model_name2,
    "Displacement", displacement2,
    "Company", company2,
    "Price", price2,
    "Cooling System", coolAir2,
    coolOil2,
    coolLiquid2
};

try {
    int rowCount = jTable1.getRowCount();
```

```
                int colCount = jTable1.getColumnCount();
                int nextRow = 0;
                boolean empty = false;
                do {
                    if (jTable1.getValueAt(nextRow, 0) != null) {
                        nextRow++;
                    } else {
                        empty = true;
                    }
                } while (empty = false && nextRow < rowCount);
                for (int i = 0; i < rowCount; i++) {
                    String model = jTable1.getValueAt(i, 0).toString().toLowerCase();
                    if (model_num2.equals(model)) {
                         int b = JOptionPane.showConfirmDialog(null, update2, " Update a
product", JOptionPane.OK_CANCEL_OPTION);
                        System.out.println(b);
                        if (b == JOptionPane.OK_OPTION) {
                            String cooling3;
                            if (coolAir2.isSelected()) {
                                cooling3 = coolAir2.getText();
                            } else if (coolLiquid2.isSelected()) {
                                cooling3 = coolLiquid2.getText();
                            } else {
                                cooling3 = coolOil2.getText();
                            }
```

```
                    String[] bikeDetails = {model_num2, model_name2.getText(), company2.getSelectedItem().toString(), displacement2.getSelectedItem().toString(), cooling3, price2.getText()};


                    for (int j = 0; j < colCount; j++) {

                        jTable1.setValueAt(bikeDetails[j], i, j);

                    }

                    JOptionPane.showMessageDialog(rootPane, model_num2 + " was successfully updated!", " Sucessfully Updated!", JOptionPane.INFORMATION_MESSAGE, icon_success);

                    loop = true;

                } else if (b == JOptionPane.CANCEL_OPTION) {

                    JOptionPane.showMessageDialog(rootPane, "The Operation was cancelled.", "Cancel", JOptionPane.WARNING_MESSAGE);

                    loop = true;

                    break;

                }

            }

            if (i == rowCount - 1 && !model_num2.equals(model)) {

                JOptionPane.showMessageDialog(rootPane, "Please enter the correct bike model bumber", "Model number not found", JOptionPane.ERROR_MESSAGE);


            }

        }

    } catch (NullPointerException e) {

    }
```

```
                } else {
                    JOptionPane.showMessageDialog(rootPane, "Please enter the model bumber of the bike and try again", "Model number not found", JOptionPane.ERROR_MESSAGE);


                }
            } else if (a == JOptionPane.CANCEL_OPTION) {
                JOptionPane.showMessageDialog(rootPane, "The Operation was cancelled.", "Model number not found", JOptionPane.WARNING_MESSAGE);
                break;
            }
        } catch (NullPointerException e) {
        };
    }



}


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


private void delete1ActionPerformed(java.awt.event.ActionEvent evt) {
    String model_num = "";
    boolean loop = false;
```

```
    while (loop == false) {
        try {
            JTextField model_num3 = new JTextField();
            Object delete[] = {"Enter the model Number", model_num3};
            int a = JOptionPane.showConfirmDialog(rootPane, delete, "Delete",
JOptionPane.OK_CANCEL_OPTION);
            model_num = model_num3.getText().toLowerCase();

            if (a == JOptionPane.OK_OPTION) {
                if (!model_num.equals("")) {

                    try {
                        int rowCount = jTable1.getRowCount();
                        int colCount = jTable1.getColumnCount();
                        int nextRow = 0;
                        boolean empty = false;
                        do {
                            if (jTable1.getValueAt(nextRow, 0) != null) {
                                nextRow++;
                            } else {
                                empty = true;
                            }
                        } while (empty = false && nextRow < rowCount);
                        for (int i = 0; i < rowCount; i++) {
                            String model = jTable1.getValueAt(i, 0).toString().toLowerCase();
```

```
                    if (model_num.equals(model))  {
                        DefaultTableModel model1 = (DefaultTableModel) jTable1.getModel();
                        model1.removeRow(i);
                        JOptionPane.showMessageDialog(rootPane, model_num + " successfully removed from the list!", " Sucessfully removed!", JOptionPane.INFORMATION_MESSAGE, icon_success);
                        loop = true;
                        break;

                    }

                    if (i == rowCount - 1) {
                        JOptionPane.showMessageDialog(rootPane, "Please enter the correct bike model bumber", "Model number not found", JOptionPane.ERROR_MESSAGE);
                        loop = false;

                    }
                }

            } catch (NullPointerException e) {
            }
        } else {
            JOptionPane.showMessageDialog(rootPane, "Please enter the model number of the bike", "Model number not found", JOptionPane.ERROR_MESSAGE);
```

```
                loop = false;


            }
        } else if (a == JOptionPane.CANCEL_OPTION) {
            JOptionPane.showMessageDialog(rootPane, "The Operation was cancelled.", "Model number not found", JOptionPane.WARNING_MESSAGE);
            loop = true;
        }
    } catch (NullPointerException e) {
    }
  }


}


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


StopWatch sw=new StopWatch();
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {


  if (!searchBar.getText().toString().equals("")) {
    try {
      int searches = Integer.parseInt(searchBar.getText().toString());
```

```
int rowCount = jTable1.getRowCount();

int colCount = jTable1.getColumnCount();


String[][] a = new String[rowCount][6];

for (int i = 0; i < a.length; i++) {

    String[] c = {jTable1.getValueAt(i, 0).toString(), jTable1.getValueAt(i, 1).toString(), jTable1.getValueAt(i, 2).toString(), jTable1.getValueAt(i, 3).toString(), jTable1.getValueAt(i, 4).toString(), jTable1.getValueAt(i, 5).toString()};

    a[i] = c;

}


sw.start();

MergeSorter.sort(a);


System.out.println("time: u"+sw.getElapsedTime());

int mid = a.length / 2;

int low = 0;

int high = 0;

if (Integer.parseInt(a[mid - 1][5]) == searches) {

    String[][] bike = new String[jTable1.getRowCount()][6];


    String[] bike1 = {"Model number: " + a[mid - 1][0], "Model Name: " + a[mid - 1][1], "Company: " + a[mid - 1][2], "Displacement: " + a[mid - 1][3], "Cooling System: " + a[mid - 1][4], "Price: " + a[mid - 1][5], "**********"};


    bike[0] = bike1;
```

```
            for (int i = 1; i < jTable1.getRowCount()  - 1; i++) {


                if (Integer.parseInt(a[mid - 1 - i][5]) == searches) {


                    String[] bike2 = {"Model number: " + a[mid - 1 - i][0], "Model Name: " +
a[mid - 1 - i][1], "Company: " + a[mid - 1 - i][2], "Displacement: " + a[mid - 1 - i][3], "Cooling
System: " + a[mid - 1 - i][4], "Price: " + a[mid - 1 - i][5], "**********************************"};

                    bike[i] = bike2;



                }

                if (Integer.parseInt(a[mid + i][5]) == searches) {


                    String[] bike2 = {"Model number: " + a[mid - 1 + i][0], "Model Name: " +
a[mid - 1 + i][1], "Company:  " + a[mid - 1 + i][2], "Displacement:  " + a[mid - 1 + i][3],
"Cooling System: " + a[mid - 1 + i][4], "Price: " + a[mid - 1 + i][5],
"**********************************"};

                    bike[i + 1] = bike2;

                }

                if (Integer.parseInt(a[mid + i][5]) != searches || Integer.parseInt(a[mid +
i][5]) != searches) {

                    break;

                }

            }

            System.out.println("Hello");


             JOptionPane.showMessageDialog(null, bike, " Found by price!",
JOptionPane.INFORMATION_MESSAGE);
```

```
            } else {
                if (searches < Integer.parseInt(a[mid - 1][5])) {
                    low = 0;
                    high = mid - 1;
                } else if (searches > Integer.parseInt(a[mid - 1][5])) {
                    low = mid - 1;
                    high = a.length;
                    System.out.println("greater than");
                }

                MotoRacers mr = new MotoRacers();
                mr.search(a, low, high, searches);
                sw.stop();

            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(rootPane, "Please enter the price to be searched in number", "Search data error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(rootPane, "Please enter the price to be searched", "Searching price not found!", JOptionPane.ERROR_MESSAGE);
    }
```

```
        JOptionPane.showMessageDialog(null,"Found in: "+sw.getElapsedTime()+"milliseconds","Time Elapsed!", JOptionPane.INFORMATION_MESSAGE);

    }


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

        try {


            if ((new File("file\\2018-19 A CS5004NI A1 CW Group Coursework L2C6 Ashutosh Chauhan 17030976.pdf")).exists()) {


                    Process p = Runtime

                      .getRuntime()

                      .exec("rundll32 url.dll,FileProtocolHandler file\\2018-19 A CS5004NI A1 CW Group Coursework L2C6 Ashutosh Chauhan 17030976.pdf");

                    p.waitFor();


            } else {


                    System.out.println("File is not exists");


            }





              } catch (Exception ex) {
```

```
        ex.printStackTrace();
 }
  }

  private void jMenu1ActionPerformed(java.awt.event.ActionEvent evt) {
      System.exit(0);
  }

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

  private void jMenu2ActionPerformed(java.awt.event.ActionEvent evt) {

  }

  private void menuHelpActionPerformed(java.awt.event.ActionEvent evt) {
      try {

        if ((new File("file\\user_guide.pdf")).exists()) {

                Process p = Runtime
                  .getRuntime()
                  .exec("rundll32 url.dll,FileProtocolHandler file\\user_guide.pdf");
```

```
                p.waitFor();

        } else {

                System.out.println("File is not exists");

        }



          } catch (Exception ex) {
        ex.printStackTrace();
 }
  }

  private void itemexitActionPerformed(java.awt.event.ActionEvent evt) {
     System.exit(0);       // TODO add your handling code here:
  }

  protected void filter() {
     int value=companyCmb.getSelectedIndex();
     String value1=companyCmb.getSelectedItem().toString();
     try{
        TableRowSorter<DefaultTableModel> sorter =new
TableRowSorter<DefaultTableModel> ((DefaultTableModel) jTable1.getModel());
```

```
            jTable1.setRowSorter(sorter);
            if(value!=0){
                sorter.setRowFilter(RowFilter.regexFilter(value1,2));
                JOptionPane.showMessageDialog(null,"Found by "+value1,"Found by
Company!", JOptionPane.INFORMATION_MESSAGE);
            }




        }
        catch(PatternSyntaxException e){};
        //catch(NullPointerException e){};


    }
        public int search(String[][] a, int low, int high, int value) {
        sw.start();
        if (low <= high) {
            System.out.println("less than");
            int mid = (low + high) / 2;
            int price1 = 0;
            try {
                price1 = Integer.parseInt(a[mid][5]);
            } catch (ArrayIndexOutOfBoundsException e) {
            }
            if (price1 == value) {
                System.out.println("less than");
```

```
String[][] bike = new String[jTable1.getRowCount()][6];


String[] bike1 = {"Model number: " + a[mid][0], "Model Name: " + a[mid][1], "Company: " + a[mid][2], "Displacement: " + a[mid][3], "Cooling System: " + a[mid][4], "Price: " + a[mid][5], "**********"};

System.out.println(Arrays.toString(bike1));

bike[0] = bike1;

for (int i = 1; i < jTable1.getRowCount() - 1; i++) {

    try {

        if (Integer.parseInt(a[mid - i][5]) == value) {

            System.out.println("Hello -id");

            String[] bike2 = {"Model number: " + a[mid - i][0], "Model Name: " + a[mid - i][1], "Company: " + a[mid - i][2], "Displacement: " + a[mid - i][3], "Cooling System: " + a[mid - i][4], "Price: " + a[mid - i][5], "**********"};

            bike[i] = bike2;



        }



        if (Integer.parseInt(a[mid + i][5]) == value) {

            System.out.println("Hello +id");

            String[] bike2 = {"Model number: " + a[mid + i][0], "Model Name: " + a[mid + i][1], "Company: " + a[mid + i][2], "Displacement: " + a[mid + i][3], "Cooling System: " + a[mid + i][4], "Price: " + a[mid + i][5], "**********"};

            bike[i + 1] = bike2;

        }
```

```
                if (Integer.parseInt(a[mid + i][5]) != value || Integer.parseInt(a[mid + i][5])
!= value) {
                    break;
                }
            } catch (ArrayIndexOutOfBoundsException e) {
            };
        }

        JOptionPane.showMessageDialog(null, bike, " Found by price!",
JOptionPane.INFORMATION_MESSAGE);

        return mid;
      } else if (price1 < value) {
        return search(a, mid + 1, high, value);
      } else {
        return search(a, low, mid - 1, value);
      }
    } else {
      JOptionPane.showMessageDialog(null, "No bike of price " + value + " was found",
" Found by price!", JOptionPane.ERROR_MESSAGE);
      return -1;
    }

  }

  /**
```

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

```
java.util.logging.Logger.getLogger(MotoRacers.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

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



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

    // Variables declaration - do not modify
    private javax.swing.JButton btnAdd;
    private javax.swing.JButton btnClear;
    private javax.swing.JButton btnExit;
    private javax.swing.JButton btnSearch;
    private javax.swing.JButton btnUpdate;
    private javax.swing.JComboBox<String> company1;
```

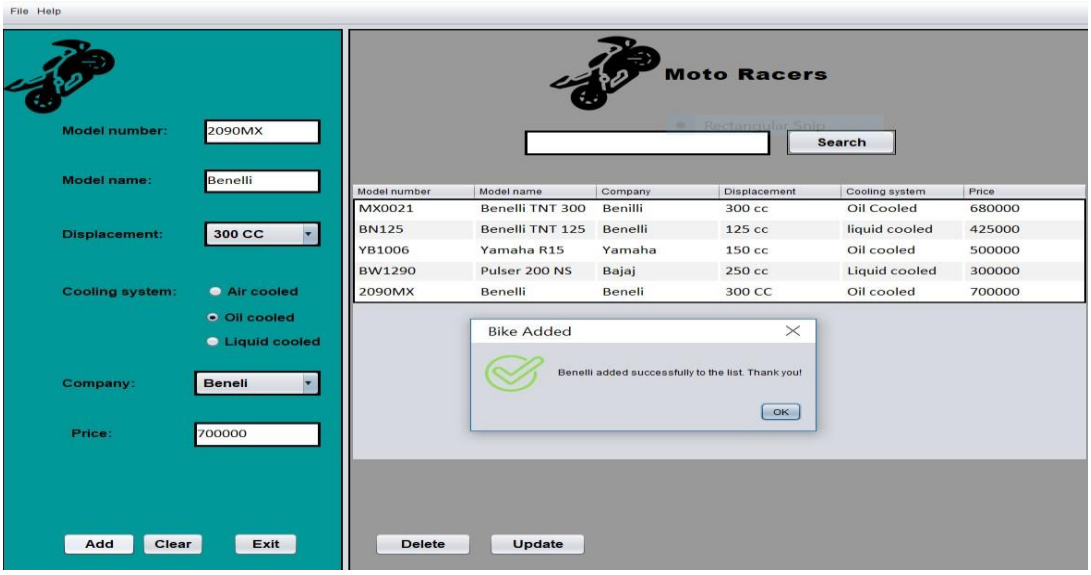
```
private javax.swing.JComboBox<String>  companyCmb;
private javax.swing.JRadioButton  coolAir;
private javax.swing.JRadioButton  coolLiquid;
private javax.swing.JRadioButton  coolOil;
private javax.swing.ButtonGroup  cooling;
private javax.swing.JButton  delete1;
private javax.swing.JComboBox<String>  displacement;
private javax.swing.JFileChooser  fileChooser;
private javax.swing.JMenuItem  itemOpen;
private javax.swing.JMenuItem  itemexit;
private javax.swing.JLabel  jLabel1;
private javax.swing.JLabel  jLabel2;
private javax.swing.JLabel  jLabel3;
private javax.swing.JLabel  jLabel4;
private javax.swing.JLabel  jLabel5;
private javax.swing.JLabel  jLabel6;
private javax.swing.JLabel  jLabel7;
private javax.swing.JLabel  jLabel8;
private javax.swing.JLabel  jLabel9;
private javax.swing.JMenu  jMenu1;
private javax.swing.JMenu  jMenu2;
private javax.swing.JMenuBar  jMenuBar1;
private javax.swing.JPanel  jPanel1;
private javax.swing.JPanel  jPanel3;
```

```
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JMenuItem menuHelp;
    private javax.swing.JTextField model_name;
    private javax.swing.JTextField model_num;
    private javax.swing.JTextField price;
    private javax.swing.JTextField searchBar;
    // End of variables declaration
}
```

9. User guide

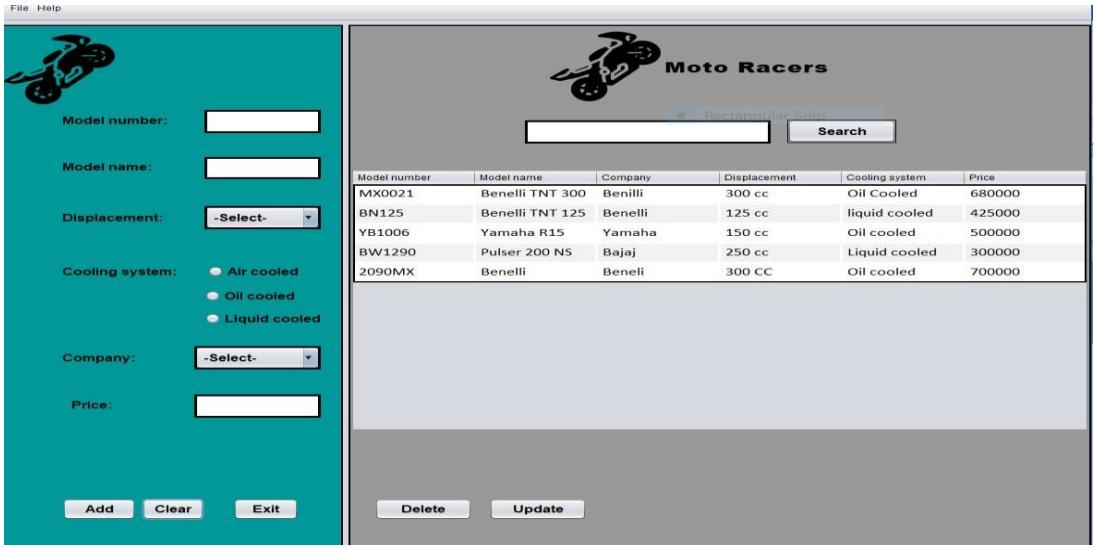
Know what each button is used for…

- Add button



The add button adds the bike detail that you enter in the form to the table on the right.

- Clear button:

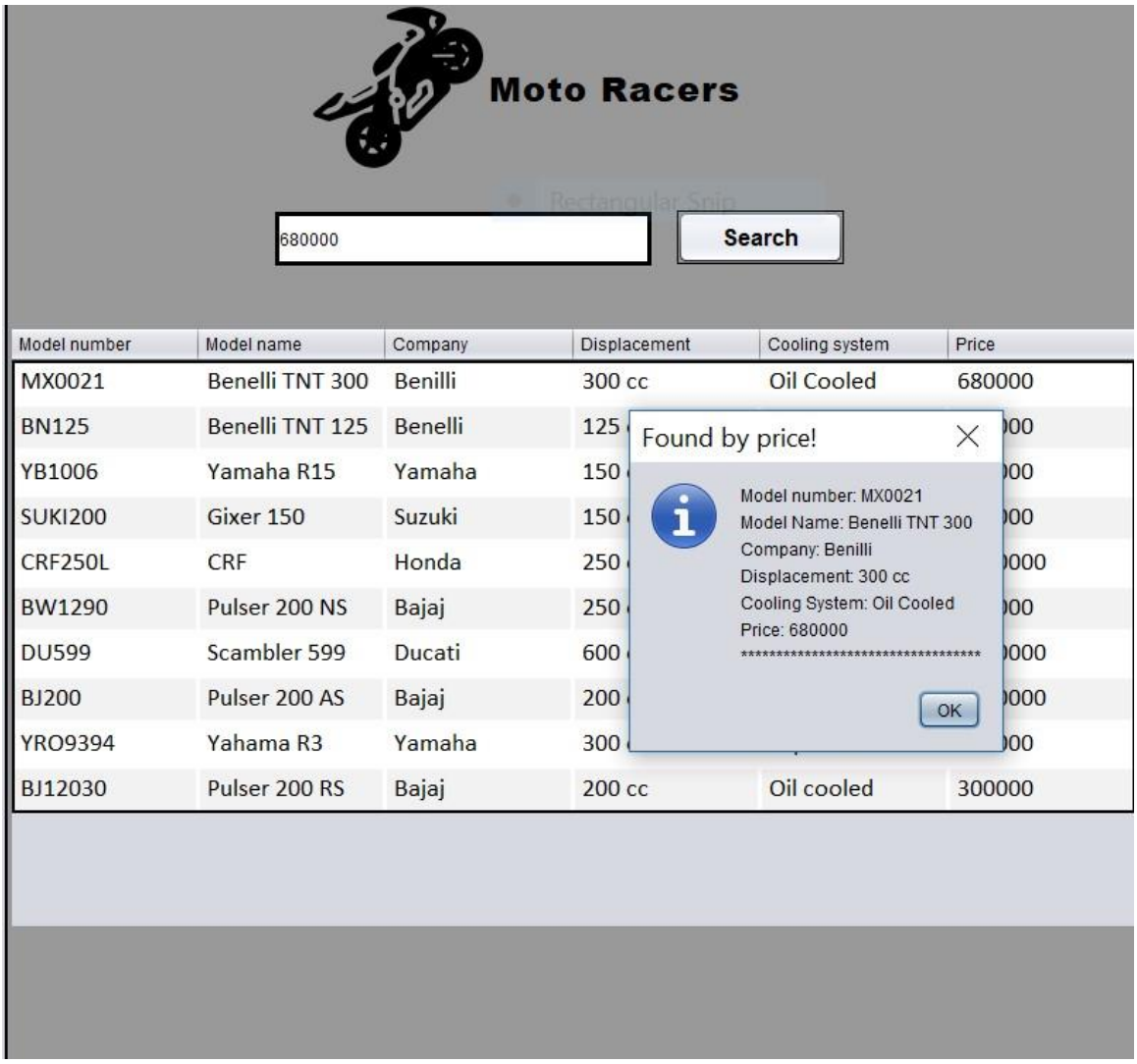


The clear button clears the values entered on the form.

- Exit:

  The exit button exits the program.

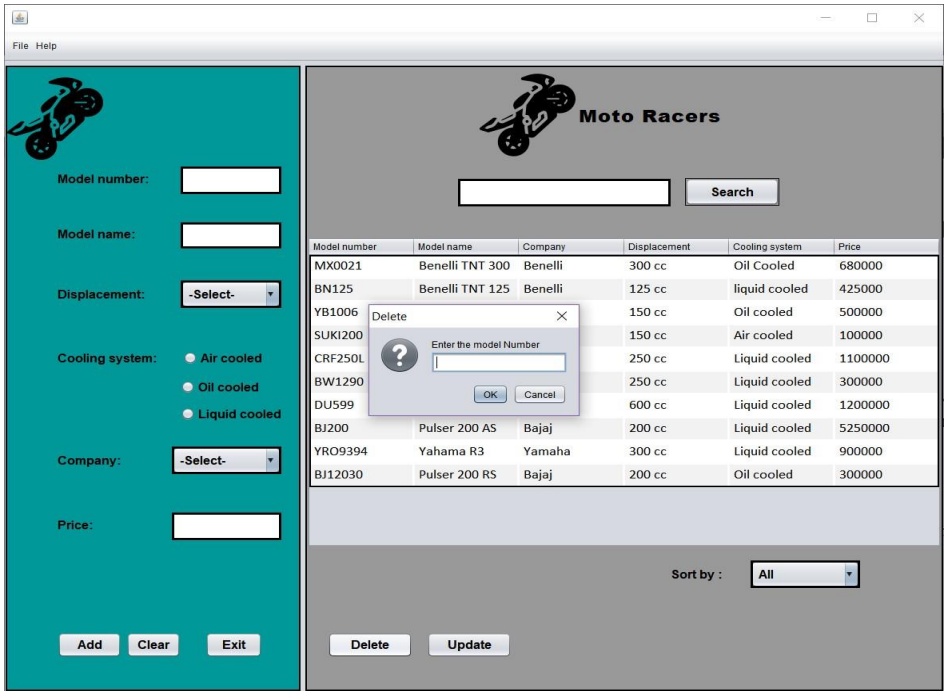
- Search:

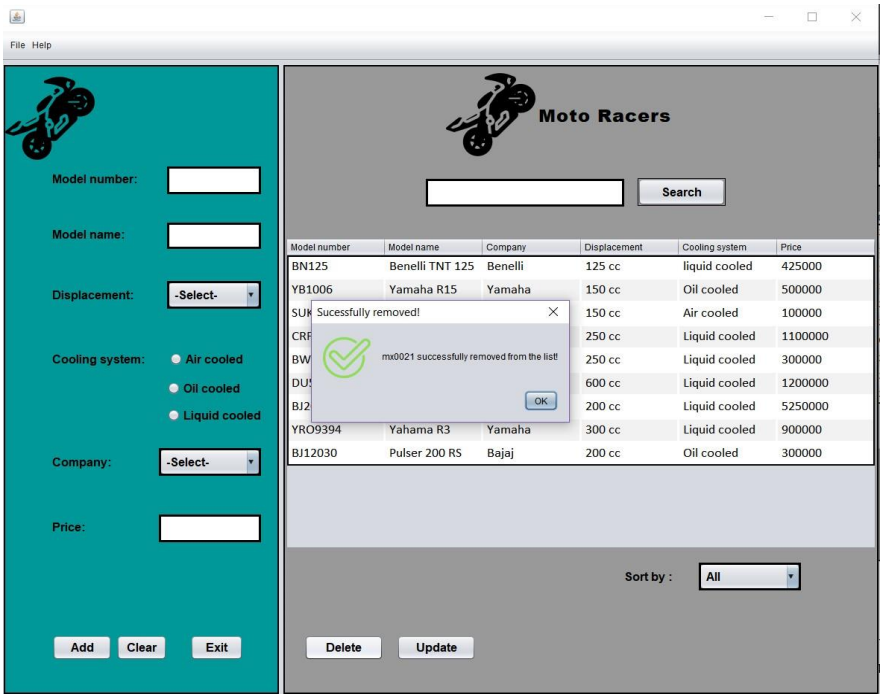


The search button searches the details of a bike according to the price entered in the search box.

- Delete:

The delete button deletes the data of the bike that you want to remove from the table. You just have to enter the model number of the bike that you want to remove from the table.


File Help
Moto Racers
Search
Model number:
Model name:
Displacement:
-Select-
Cooling system:
Air cooled
Oil cooled
Liquid cooled
Company:
-Select-
Price:
Add
Clear
Exit
Delete
Update
Sort by :
All
Model number
Model name
Company
Displacement
Cooling system
Price
MX0021 Benelli TNT 300 Benelli 300 cc Oil Cooled 680000
BN125 Benelli TNT 125 Benelli 125 cc liquid cooled 425000
YB1006 150 cc Oil cooled 500000
SUKI200 150 cc Air cooled 100000
CRF250L 250 cc Liquid cooled 1100000
BW1290 250 cc Liquid cooled 300000
DU599 600 cc Liquid cooled 1200000
BJ200 Pulser 200 AS Bajaj 200 cc Liquid cooled 5250000
YRO9394 Yahama R3 Yamaha 300 cc Liquid cooled 900000
BJ12030 Pulser 200 RS Bajaj 200 cc Oil cooled 300000
Delete
Enter the model Number
OK
Cancel


File Help
Moto Racers
Search
Model number:
Model name:
Displacement:
-Select-
Cooling system:
Air cooled
Oil cooled
Liquid cooled
Company:
-Select-
Price:
Add
Clear
Exit
Delete
Update
Sort by :
All
Model number
Model name
Company
Displacement
Cooling system
Price
BN125 Benelli TNT 125 Benelli 125 cc liquid cooled 425000
YB1006 Yamaha R15 Yamaha 150 cc Oil cooled 500000
150 cc Air cooled 100000
250 cc Liquid cooled 1100000
250 cc Liquid cooled 300000
600 cc Liquid cooled 1200000
200 cc Liquid cooled 5250000
YRO9394 Yahama R3 Yamaha 300 cc Liquid cooled 900000
BJ12030 Pulser 200 RS Bajaj 200 cc Oil cooled 300000
Sucessfully removed!
mx0021 successfully removed from the list!
OK

- Update:

The update button updates the data of the bike that you want to update. You just have to enter the model number of the bike that you want to update.