10/7/2019

DATA MINING
(CS 634-META-19F)

MIDTERM PROJECT -
Under guidance of
Professor Jason Wang

Submitted By -

Name - Garima Tuli

Email Id -

gt98@njit.edu

Table of Contents

Platforms Used

Programming Language – Java

Development Environment – Eclipse IDE

Operating System – Windows

Hardware – Laptop

Instructions for Running the Program

- This Apriori Implementation is written in Java. To run it, right click on the java file, Click "Run as Java Application"
- You need to change the Dataset name in the program code each time you need to run the program on a different dataset as shown below:

```
String fileName = "dataset\\database1.txt";
String fileName = "dataset\\database2.txt";
String fileName = "dataset\\database3.txt";
String fileName = "dataset\\database4.txt";
String fileName = "dataset\\database5.txt";
```

- Program asks user to enter the minimum support value and minimum confidence value. After user gives input, `the association Rules satisfying the minimum Support & Confidence Values entered by user gets generated in the output.`

Input (Datasets Used)

Dataset 1

- **10 Items used in Dataset_1 are as follows:**

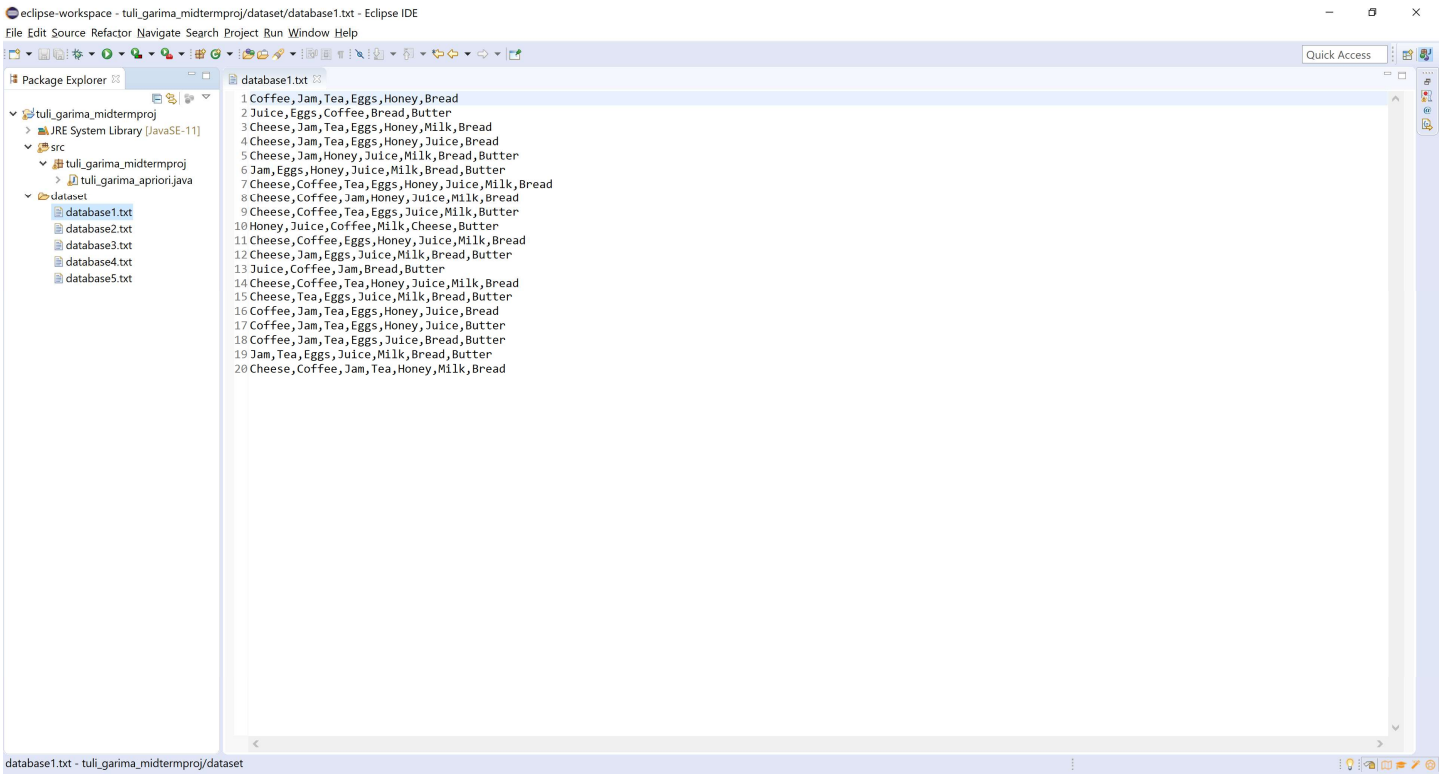
  Coffee,Jam,Tea,Eggs,Honey,Bread,Juice,Cheese,Milk,Butter

- **Transactions for above dataset are as follows:**

  Coffee,Jam,Tea,Eggs,Honey,Bread
  Juice,Eggs,Coffee,Bread,Butter
  Cheese,Jam,Tea,Eggs,Honey,Milk,Bread

Cheese,Jam,Tea,Eggs,Honey,Juice,Bread
Cheese,Jam,Honey,Juice,Milk,Bread,Butter
Jam,Eggs,Honey,Juice,Milk,Bread,Butter
Cheese,Coffee,Tea,Eggs,Honey,Juice,Milk,Bread
Cheese,Coffee,Jam,Honey,Juice,Milk,Bread
Cheese,Coffee,Tea,Eggs,Juice,Milk,Butter
Honey,Juice,Coffee,Milk,Cheese,Butter
Cheese,Coffee,Eggs,Honey,Juice,Milk,Bread
Cheese,Jam,Eggs,Juice,Milk,Bread,Butter
Juice,Coffee,Jam,Bread,Butter
Cheese,Coffee,Tea,Honey,Juice,Milk,Bread
Cheese,Tea,Eggs,Juice,Milk,Bread,Butter
Coffee,Jam,Tea,Eggs,Honey,Juice,Bread
Coffee,Jam,Tea,Eggs,Honey,Juice,Butter
Coffee,Jam,Tea,Eggs,Juice,Bread,Butter
Jam,Tea,Eggs,Juice,Milk,Bread,Butter
Cheese,Coffee,Jam,Tea,Honey,Milk,Bread

- **Screenshot of Dataset_1 from Eclipse IDE**



Dataset 2

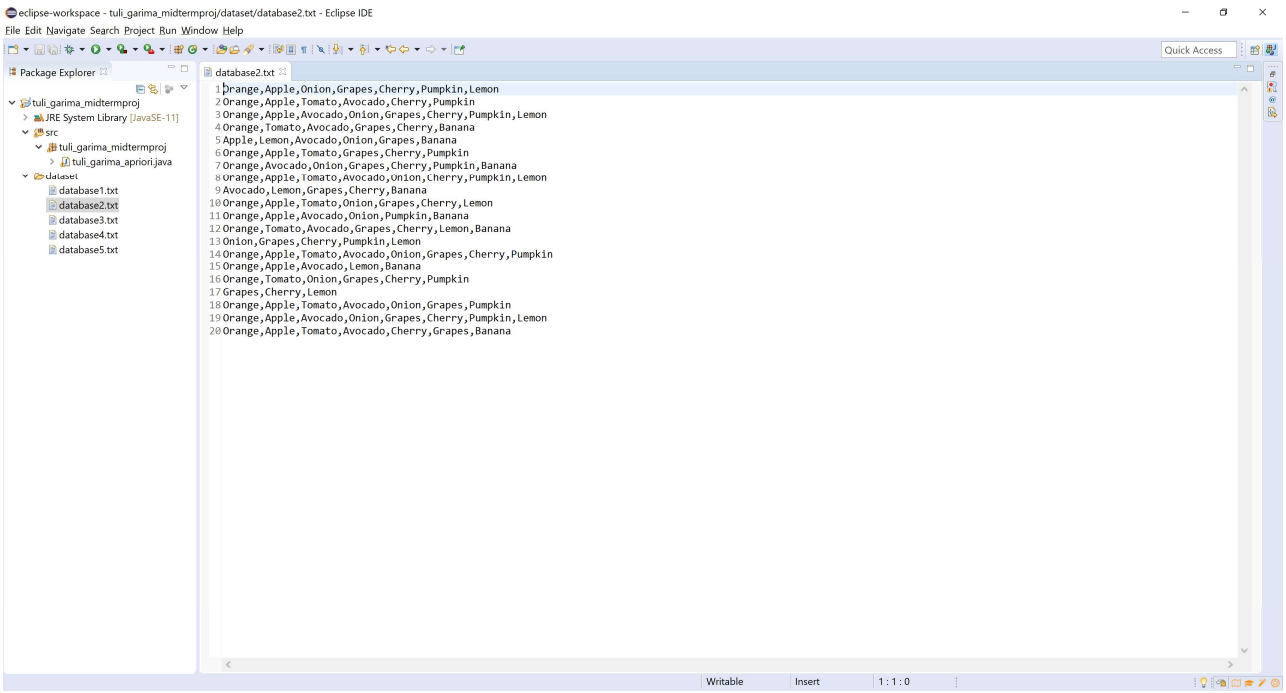
- **10 Items used in Dataset_2 are as follows:**

  Orange,Apple,Pumpkin,Cherry,Grapes,Avocado,Tomato,Onion,Lemon,Banana

- **Transactions for above dataset are as follows:**

  Orange,Apple,Onion,Grapes,Cherry,Pumpkin,Lemon
  Orange,Apple,Tomato,Avocado,Cherry,Pumpkin
  Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
  Orange,Tomato,Avocado,Grapes,Cherry,Banana
  Apple,Lemon,Avocado,Onion,Grapes,Banana
  Orange,Apple,Tomato,Grapes,Cherry,Pumpkin
  Orange,Avocado,Onion,Grapes,Cherry,Pumpkin,Banana
  Orange,Apple,Tomato,Avocado,Onion,Cherry,Pumpkin,Lemon
  Avocado,Lemon,Grapes,Cherry,Banana
  Orange,Apple,Tomato,Onion,Grapes,Cherry,Lemon
  Orange,Apple,Avocado,Onion,Pumpkin,Banana
  Orange,Tomato,Avocado,Grapes,Cherry,Lemon,Banana
  Onion,Grapes,Cherry,Pumpkin,Lemon
  Orange,Apple,Tomato,Avocado,Onion,Grapes,Cherry,Pumpkin
  Orange,Apple,Avocado,Lemon,Banana
  Orange,Tomato,Onion,Grapes,Cherry,Pumpkin
  Grapes,Cherry,Lemon
  Orange,Apple,Tomato,Avocado,Onion,Grapes,Pumpkin
  Orange,Apple,Avocado,Onion,Grapes,Cherry,Pumpkin,Lemon
  Orange,Apple,Tomato,Avocado,Cherry,Grapes,Banana

- **Screenshot of Dataset_2 from Eclipse IDE**



Dataset 3

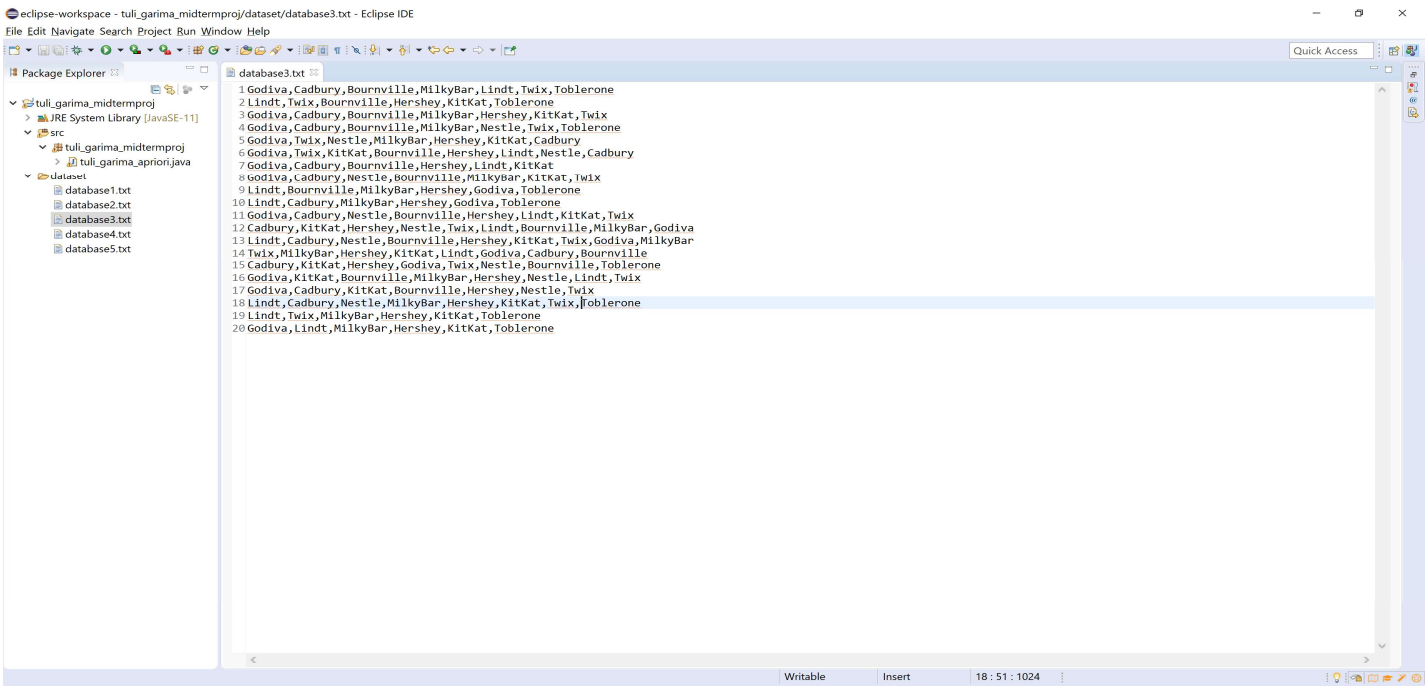
- **10 Items used in Dataset_3 are as follows:**

  Lindt,Bournville,Godiva,Twix,Cadbury,KitKat,Hershey,MilkyBar,Toblerone,Nestle

- **Transactions for above dataset are as follows:**

  Godiva,Cadbury,Bournville,MilkyBar,Lindt,Twix,Toblerone
  Lindt,Twix,Bournville,Hershey,KitKat,Toblerone
  Godiva,Cadbury,Bournville,MilkyBar,Hershey,KitKat,Twix
  Godiva,Cadbury,Bournville,MilkyBar,Nestle,Twix,Toblerone
  Godiva,Twix,Nestle,MilkyBar,Hershey,KitKat,Cadbury
  Godiva,Twix,KitKat,Bournville,Hershey,Lindt,Nestle,Cadbury
  Godiva,Cadbury,Bournville,Hershey,Lindt,KitKat
  Godiva,Cadbury,Nestle,Bournville,MilkyBar,KitKat,Twix
  Lindt,Bournville,MilkyBar,Hershey,Godiva,Toblerone
  Lindt,Cadbury,MilkyBar,Hershey,Godiva,Toblerone
  Godiva,Cadbury,Nestle,Bournville,Hershey,Lindt,KitKat,Twix
  Cadbury,KitKat,Hershey,Nestle,Twix,Lindt,Bournville,MilkyBar,Godiva
  Lindt,Cadbury,Nestle,Bournville,Hershey,KitKat,Twix,Godiva,MilkyBar
  Twix,MilkyBar,Hershey,KitKat,Lindt,Godiva,Cadbury,Bournville
  Cadbury,KitKat,Hershey,Godiva,Twix,Nestle,Bournville,Toblerone
  Godiva,KitKat,Bournville,MilkyBar,Hershey,Nestle,Lindt,Twix
  Godiva,Cadbury,KitKat,Bournville,Hershey,Nestle,Twix
  Lindt,Cadbury,Nestle,MilkyBar,Hershey,KitKat,Twix,Toblerone
  Lindt,Twix,MilkyBar,Hershey,KitKat,Toblerone
  Godiva,Lindt,MilkyBar,Hershey,KitKat,Toblerone

- **Screenshot of Dataset_3 from Eclipse IDE**



Dataset 4

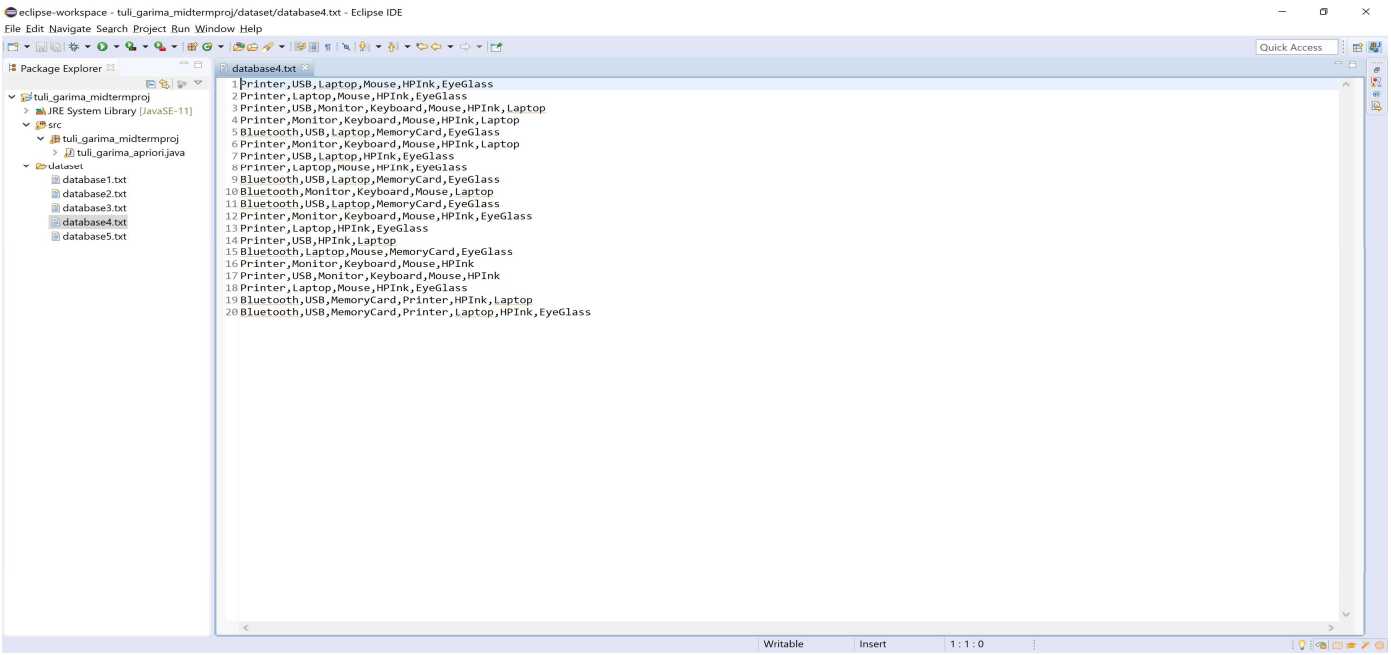
- **10 Items used in Dataset_4 are as follows:**

  Printer,Bluetooth,USB,Laptop,Monitor,Keyboard,Mouse,HPInk,MemoryCard,EyeGlass

- **Transactions for above dataset are as follows:**

  Printer,USB,Laptop,Mouse,HPInk,EyeGlass
  Printer,Laptop,Mouse,HPInk,EyeGlass
  Printer,USB,Monitor,Keyboard,Mouse,HPInk,Laptop
  Printer,Monitor,Keyboard,Mouse,HPInk,Laptop
  Bluetooth,USB,Laptop,MemoryCard,EyeGlass
  Printer,Monitor,Keyboard,Mouse,HPInk,Laptop
  Printer,USB,Laptop,HPInk,EyeGlass
  Printer,Laptop,Mouse,HPInk,EyeGlass
  Bluetooth,USB,Laptop,MemoryCard,EyeGlass
  Bluetooth,Monitor,Keyboard,Mouse,Laptop
  Bluetooth,USB,Laptop,MemoryCard,EyeGlass
  Printer,Monitor,Keyboard,Mouse,HPInk,EyeGlass
  Printer,Laptop,HPInk,EyeGlass
  Printer,USB,HPInk,Laptop
  Bluetooth,Laptop,Mouse,MemoryCard,EyeGlass
  Printer,Monitor,Keyboard,Mouse,HPInk
  Printer,USB,Monitor,Keyboard,Mouse,HPInk
  Printer,Laptop,Mouse,HPInk,EyeGlass
  Bluetooth,USB,MemoryCard,Printer,HPInk,Laptop
  Bluetooth,USB,MemoryCard,Printer,Laptop,HPInk,EyeGlass

- **Screenshot of Dataset_4 from Eclipse IDE**



Dataset 5

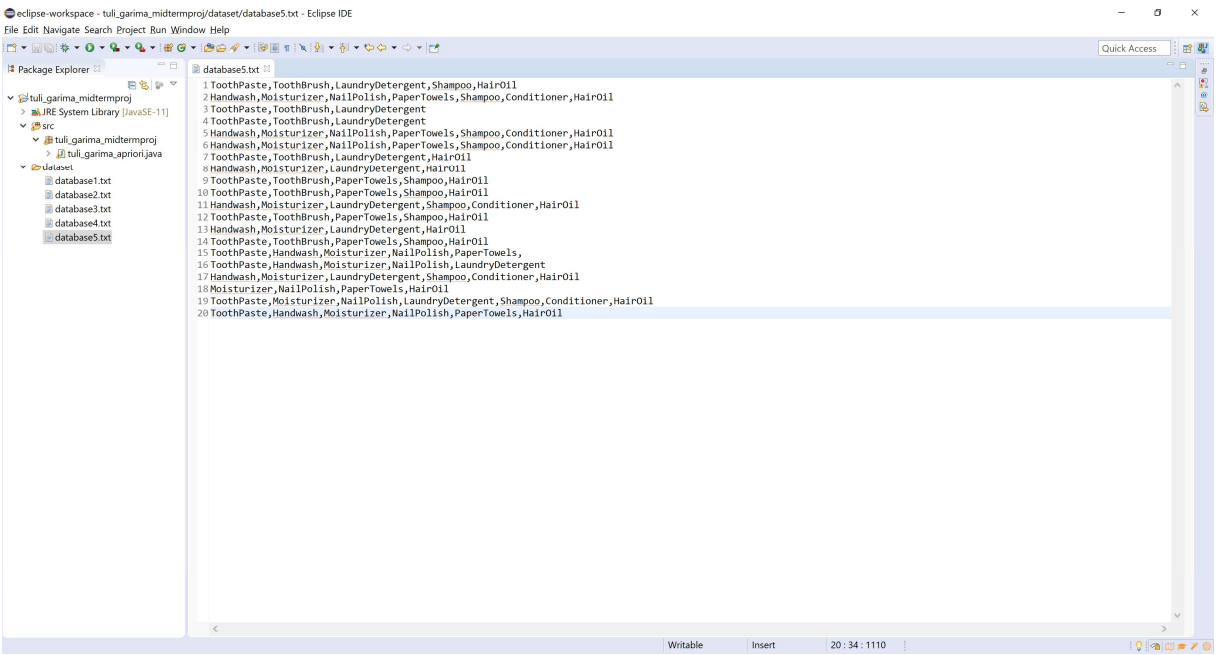
- **10 Items used in Dataset_5 are as follows:**

  ToothPaste,ToothBrush,Handwash,Moisturizer,NailPolish,LaundryDetergent,PaperTowels,Shampoo,Conditioner,HairOil

- **Transactions for above dataset are as follows:**

  ToothPaste,ToothBrush,LaundryDetergent,Shampoo,HairOil
  Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
  ToothPaste,ToothBrush,LaundryDetergent
  ToothPaste,ToothBrush,LaundryDetergent
  Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
  Handwash,Moisturizer,NailPolish,PaperTowels,Shampoo,Conditioner,HairOil
  ToothPaste,ToothBrush,LaundryDetergent,HairOil
  Handwash,Moisturizer,LaundryDetergent,HairOil
  ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
  ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
  Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
  ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
  Handwash,Moisturizer,LaundryDetergent,HairOil
  ToothPaste,ToothBrush,PaperTowels,Shampoo,HairOil
  ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,
  ToothPaste,Handwash,Moisturizer,NailPolish,LaundryDetergent
  Handwash,Moisturizer,LaundryDetergent,Shampoo,Conditioner,HairOil
  Moisturizer,NailPolish,PaperTowels,HairOil
  ToothPaste,Moisturizer,NailPolish,LaundryDetergent,Shampoo,Conditioner,HairOil
  ToothPaste,Handwash,Moisturizer,NailPolish,PaperTowels,HairOil

- **Screenshot of Dataset_5 from Eclipse IDE**



SOURCE CODE FOR APRIORI ALGORITHM IMPLEMENTATION IN JAVA

```
package tuli_garima_midtermproj;

import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class tuli_garima_apriori {

	// ArrayList to store unique items from all transactions
	private static ArrayList<String> uniqueItemsList = new ArrayList<String>();

	// Data Structures to store all transactions
	private static ArrayList<HashSet<String>> transactionAllItemsList
	= new ArrayList<HashSet<String>>();

	// HashMap to store count per item in each transaction
	private static HashMap<String, Integer> currentCandidacyList = new HashMap<String,
Integer>();

	// ArrayList to store frequent items
	private static ArrayList<String> frequentItemsList = new ArrayList<String>();

	// HashMap to store items whose count/no of transactions >= support
	private static HashMap<String, Integer> frequentItemsForAssociations
	= new HashMap<String, Integer>();

	// ArrayList to store frequent List after first iteration
	private static ArrayList<String> frequentItemsListAfterFirstIteration
	= new ArrayList<String>();

	// ArrayList to store each possible combination from frequent list
	private static ArrayList<String> combinationItemsList = new ArrayList<String>();



	// Main Function

	public static void main(String args[]) {
		System.out.println("\nWelcome to Garima Tuli's Apriori Algorithm\n");
		System.out.println("***************************");
		System.out.println("\nFollowing is the Input Dataset:");
		System.out.println();

	// Call Function to read dataset from file
		readDataFromFile();
		System.out.println("\n***************************");
		System.out.println();
```

```
// Input Minimum Support Value from User
        System.out.println("Enter Minimum Support Value: ");
        Scanner sc = new Scanner(System.in);
        double support = sc.nextDouble();
        System.out.println();

// Input Minimum Confidence Value from User
        System.out.println("Enter Minimum Confidence Value: ");
        double confidence = sc.nextDouble();
        System.out.println("\n*****************************");
        System.out.println("\nAssociation Rules satisfying the minimum Support
                           & Confidence Values entered by user above are as follows:");


// Logic to generate candidacy List, frequent item set and associations

        int itemCount = 1;

        if (itemCount == 1) {
              candidacyListforOneorAllCombinations(uniqueItemsList, itemCount);
              frequentList(support);
              frequentItemsListAfterFirstIteration
              = new  ArrayList<String>(frequentItemsList);
              itemCount = itemCount + 1;
        }

        if (itemCount == 2) {
              String[] itemCountArray = new String[itemCount];
              combinationItemsList.clear();

              ListWithTwoItems(frequentItemsListAfterFirstIteration,
              frequentItemsListAfterFirstIteration.size(),
              itemCount, 0, itemCountArray, 0);

              candidacyListforOneorAllCombinations(combinationItemsList, itemCount);
              frequentList(support);
              itemCount = itemCount + 1;
        }

        if (itemCount == 3) {
              while (frequentItemsList.size() != 0) {

                    ListWithThreeOrMoreItems(frequentItemsList, itemCount);
                    candidacyListforOneorAllCombinations(combinationItemsList,itemCount);
                    frequentList(support);
                    itemCount = itemCount + 1;
              }
        }

        findAllAssociations(frequentItemsForAssociations, confidence);
 }
```

```
	// Read Data from File

	public static void readDataFromFile() {

	// Replace different file name here for different data set
		String fileName = "dataset\\database1.txt";
	// Local variable for each line
		String eachLine = null;
	// Unique Items
		HashSet<String> uniqueItems = new HashSet<String>();

		try {

			FileReader fileReader = new FileReader(fileName);
			BufferedReader bufferedReader = new BufferedReader(fileReader);

			while ((eachLine = bufferedReader.readLine()) != null) {

				// Local Items to form transactionItemsList
				HashSet<String> localItems = new HashSet<String>();
				String formedList[] = eachLine.split(",");

				for (int i = 0; i < formedList.length; i++) {
					localItems.add(formedList[i]);
					uniqueItems.add(formedList[i]);
				}

				System.out.println(eachLine);

				// transactionItemsList will contain set of all itemset
				transactionAllItemsList.add(localItems);
			}

		} catch (Exception e) {
			System.out.println("In to Exception" + e);
		}

		// finalList is arraylist of unique items from itemset
		uniqueItemsList = new ArrayList<String>(uniqueItems);

	}




// Function to form candidacy list (for one or all possible combinations in all transactions)

		public static void candidacyListforOneorAllCombinations(ArrayList<String>
		receivedUniqueItemsList, int receivedCount) {

		 if (receivedCount == 1) {
			currentCandidacyList.clear();

		       // Loop through final List which is a unique list (finalList)
		       // and all items (transactionItemsList)
```

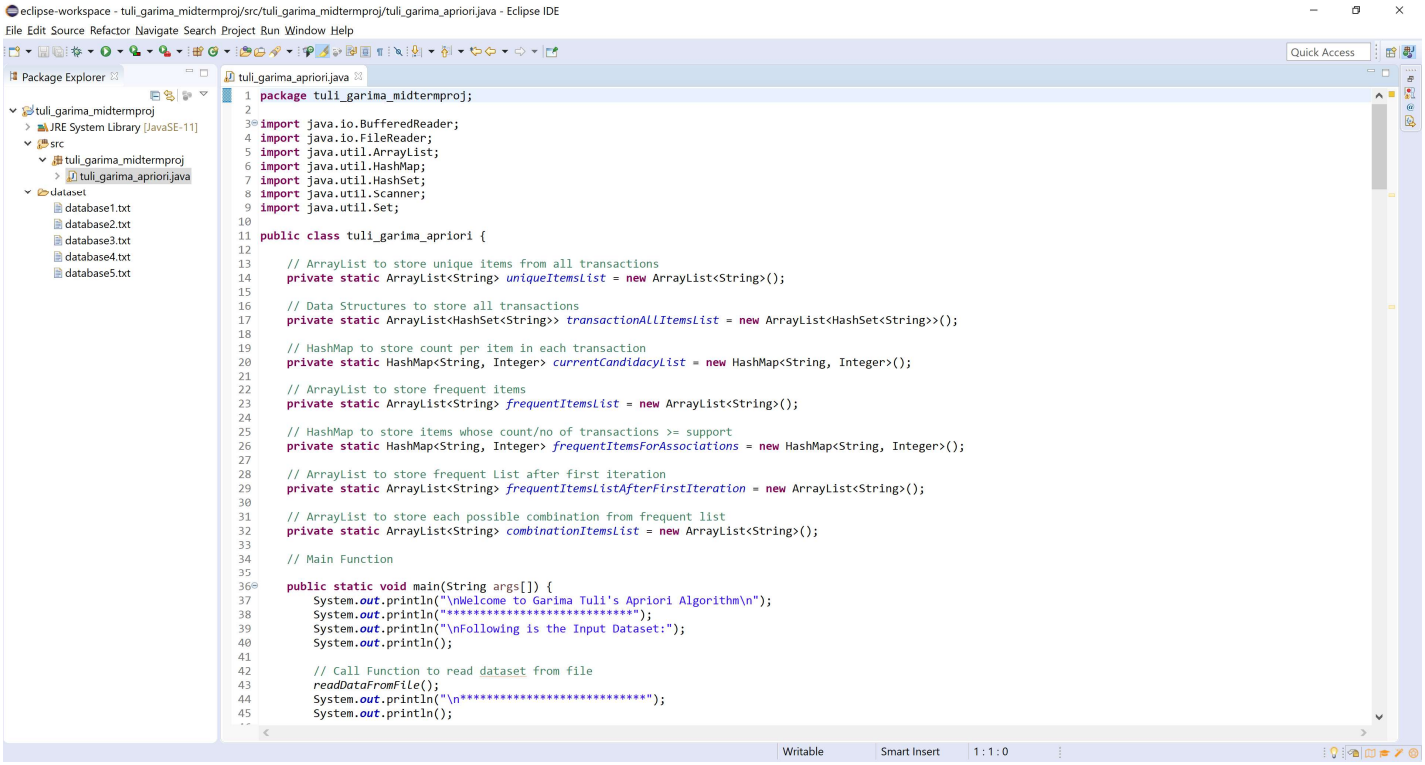
```
                // to generate count per item in each transaction

                for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
                        String oneItem = receivedUniqueItemsList.get(i);

                        for (int j = 0; j < transactionAllItemsList.size(); j++) {
                                if (transactionAllItemsList.get(j).contains(oneItem)) {
                                        if (currentCandidacyList.containsKey(oneItem)) {
                                                int counter = currentCandidacyList.get(oneItem);
                                                currentCandidacyList.put(oneItem, ++counter);
                                        } else {
                                                currentCandidacyList.put(oneItem, 1);
                                        }
                                }
                        }
                }
        } else {
                currentCandidacyList.clear();
                for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
                        String st = receivedUniqueItemsList.get(i);
                        String[] splitString = st.split(" ");
                        for (int j = 0; j < transactionAllItemsList.size(); j++) {
                                boolean fl = true;
                                for (int k = 0; k < splitString.length; k++) {

                                if(!transactionAllItemsList.get(j).contains(splitString[k]))
                                 {
                                        fl = false;
                                        break;
                                }
                                }
                                if (fl == true) {
                                        if (currentCandidacyList.containsKey(st)) {
                                                int value = currentCandidacyList.get(st);
                                                currentCandidacyList.put(st, ++value);
                                        } else {
                                                currentCandidacyList.put(st, 1);
                                        }
                                }
                        }
                }
        }
}



 // Function to form ArrayList for two items per transaction

public static void ListWithTwoItems(ArrayList<String> receivedFrequentItemsList,
 int receivedFrequentItemsListSize,int itemCountSize, int i,
 String[] itemCountArray, int j) {

  if (i == itemCountSize) {
                String st = "";
                for (int k = 0; k < itemCountSize; k++) {
```

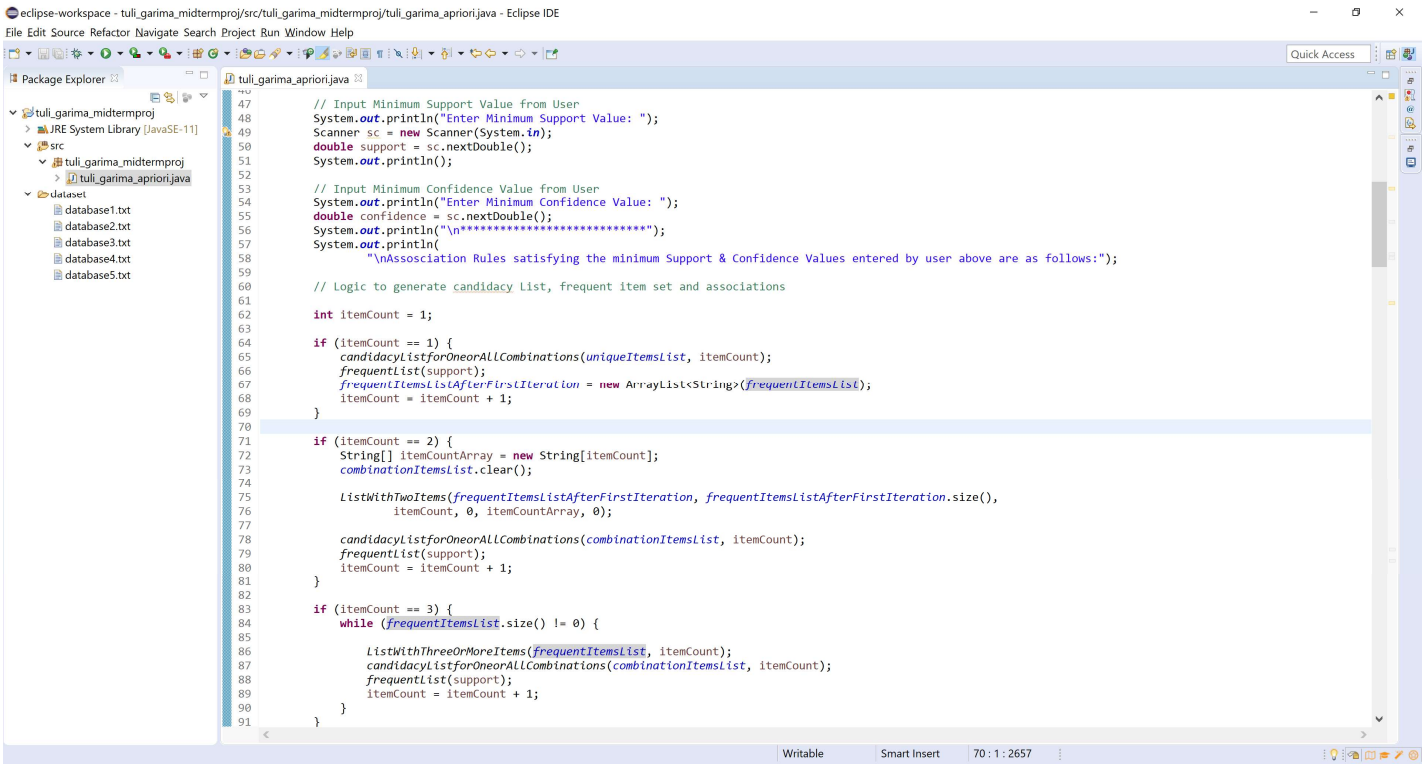
```
                        st = st + itemCountArray[k] + " ";
                }
                combinationItemsList.add(st);
                return;
        }
        if (j >= receivedFrequentItemsListSize) {
                return;
        }

        itemCountArray[i] = receivedFrequentItemsList.get(j);

        ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize,
        itemCountSize, i + 1, itemCountArray,j + 1);

        ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize,
        itemCountSize, i, itemCountArray,j + 1);

}



 // Function to form ArrayList for three or more items per transaction

private static void ListWithThreeOrMoreItems(ArrayList<String> receivedFrequentItemsList,
int receivedCount) {
       combinationItemsList.clear();

       for (int i = 0; i < receivedFrequentItemsList.size(); i++) {
              String eachString = receivedFrequentItemsList.get(i).trim();
              String st = eachString.substring(0, eachString.lastIndexOf(" "));

              for (int j = i + 1; j < receivedFrequentItemsList.size(); j++) {
                     String eachString1 = receivedFrequentItemsList.get(j).trim();
                     String st1 = eachString1.substring(0, eachString1.lastIndexOf(" "));
                     if (st.equals(st1)) {
                     combinationItemsList.add(eachString + " " +
                     eachString1.substring(eachString1.lastIndexOf(" ") + 1));
                     }
              }
       }
}




      // Function to form frequent List and frequent List for association
       public static void frequentList(double receivedSupport) {
       frequentItemsList.clear();
       receivedSupport = receivedSupport / 100;
       Set<String> items = currentCandidacyList.keySet();

       for (String eachItem : items) {
              if ((double) currentCandidacyList.get(eachItem)
                   / transactionAllItemsList.size() >= receivedSupport) {
                     frequentItemsList.add(eachItem);
```

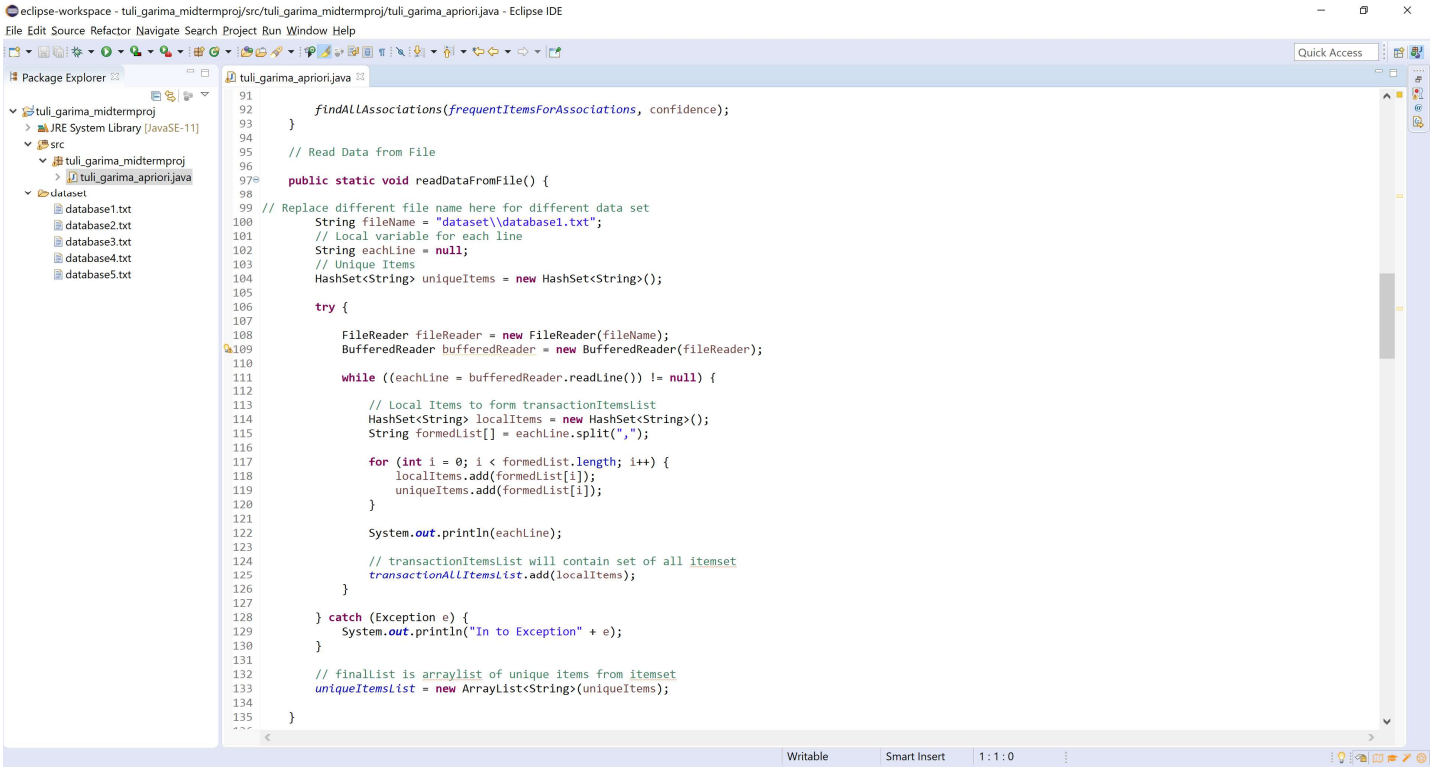
```
                        frequentItemsForAssociations.put(eachItem,
                        currentCandidacyList.get(eachItem));
                }
        }
}


        // Function to form all the associations
        public static void findAllAssociations(HashMap<String, Integer>
        receivedFrequentItemsForAssociations, double confidenceFromUser) {

         for (String st : receivedFrequentItemsForAssociations.keySet()) {
                String[] splitArray = st.split(" ");

                if (splitArray.length == 2) {
                    printAssociationBasedOnConfidence(st,
                    receivedFrequentItemsForAssociations.get(st),
                     splitArray[0],splitArray[1], confidenceFromUser);

                       printAssociationBasedOnConfidence(st,
                        receivedFrequentItemsForAssociations.get(st),
                        splitArray[1],splitArray[0], confidenceFromUser);
                } else if (splitArray.length == 3) {

                       for (int i = 0; i < splitArray.length; i++) {
                             if (i == 0) {
                                printAssociationBasedOnConfidence(st,
                                receivedFrequentItemsForAssociations.get(st),
                                splitArray[i], splitArray[i + 1] + " " +
                                splitArray[i + 2], confidenceFromUser);

                               printAssociationBasedOnConfidence(st,
                               receivedFrequentItemsForAssociations.get(st),
                               splitArray[i + 1] + " " +
                               splitArray[i + 2], splitArray[i], confidenceFromUser);
                             } else if (i == 1) {

                               printAssociationBasedOnConfidence(st,
                               receivedFrequentItemsForAssociations.get(st),
                               splitArray[i], splitArray[i - 1] + " " +
                               splitArray[i + 1], confidenceFromUser);

                               printAssociationBasedOnConfidence(st,
                               receivedFrequentItemsForAssociations.get(st),
                               splitArray[i - 1] + " " + splitArray[i + 1],
                               splitArray[i], confidenceFromUser);
                             } else {
                                 printAssociationBasedOnConfidence(st,
                                 receivedFrequentItemsForAssociations.get(st),
                                 splitArray[i], splitArray[i - 2] + " " +
                                 splitArray[i - 1], confidenceFromUser);

                                 printAssociationBasedOnConfidence(st,
                                 receivedFrequentItemsForAssociations.get(st),
                                 splitArray[i - 2] + " " + splitArray[i - 1],
                                 splitArray[i], confidenceFromUser);
```

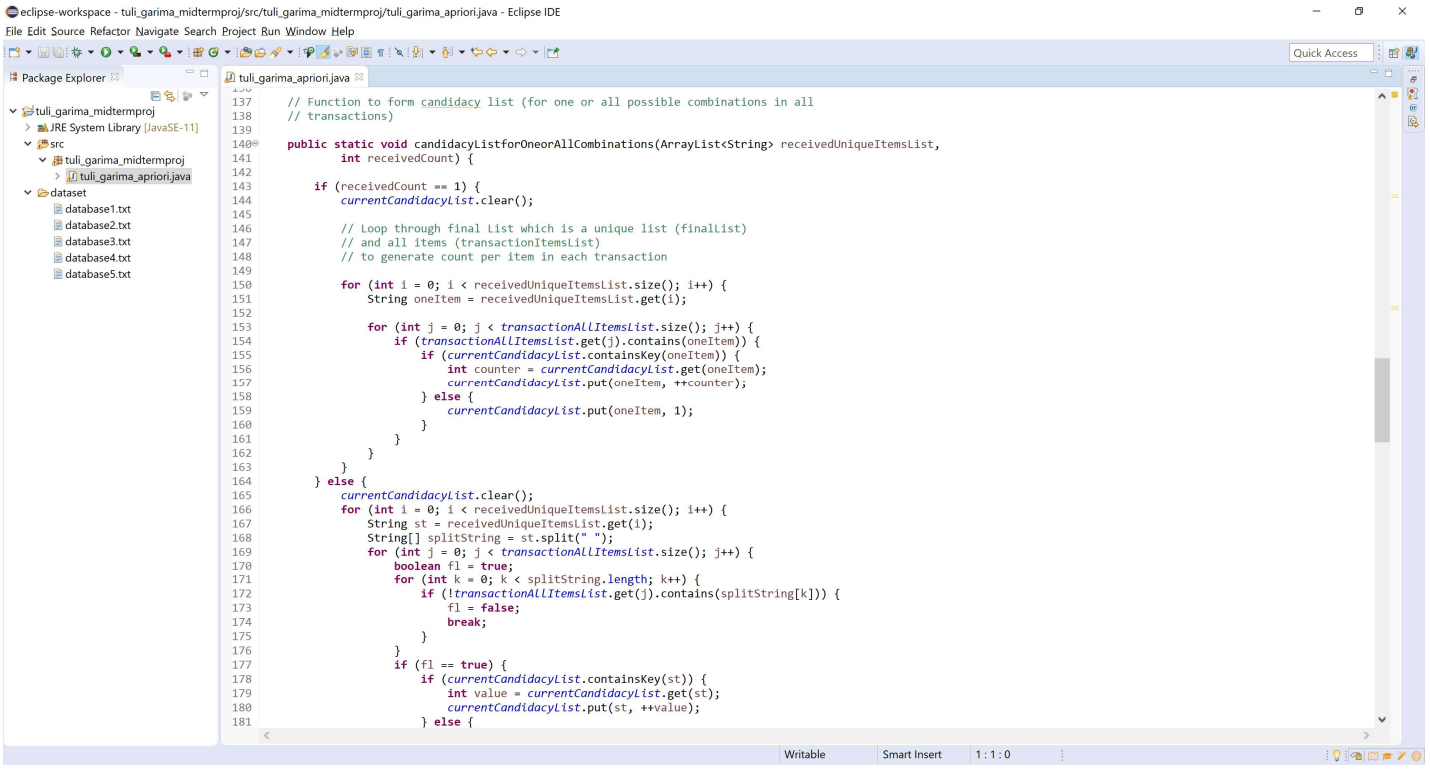
```
                                    }
                              }
                        }
                  }
            }


// Function to print associations for confidence greater than user provided minimum confidence
value

            public static void printAssociationBasedOnConfidence(String eachFrequentListItem,
            int countOfEachFrequentListItem,String leftHandSide,
            String rightHandSide, double confidenceFromUser) {

                        int count = 0;
                        String[] splitArray = leftHandSide.split(" ");
                        int length = splitArray.length;

                        for (int i = 0; i < transactionAllItemsList.size(); i++) {
                              boolean fl = true;
                              for (int j = 0; j < length; j++) {
                        if(!transactionAllItemsList.get(i).contains(splitArray[j]))
                                    {
                                          fl = false;
                                          break;
                                     }
                              }
                              if (fl == true) {
                                    count = count + 1;
                              }
                        }

            double confidenceOfEachAssociation
            = (double) countOfEachFrequentListItem / (count * 1.0);

            double percentageUserConfidence = (double) confidenceFromUser / 100;

            if (confidenceOfEachAssociation >= percentageUserConfidence) {
                  System.out.println(leftHandSide + "->" + rightHandSide);
            }
      }

}
```

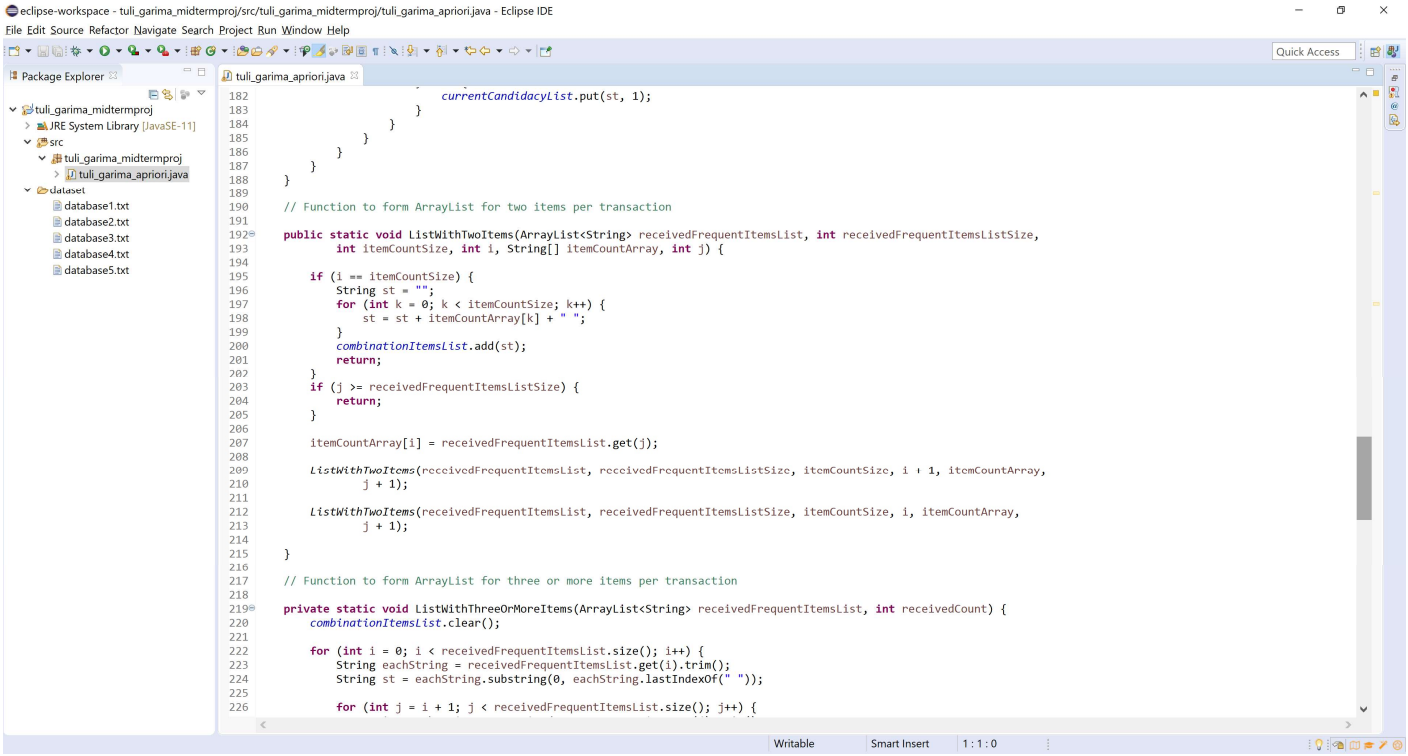
Screenshot of Source Code from Eclipse IDE

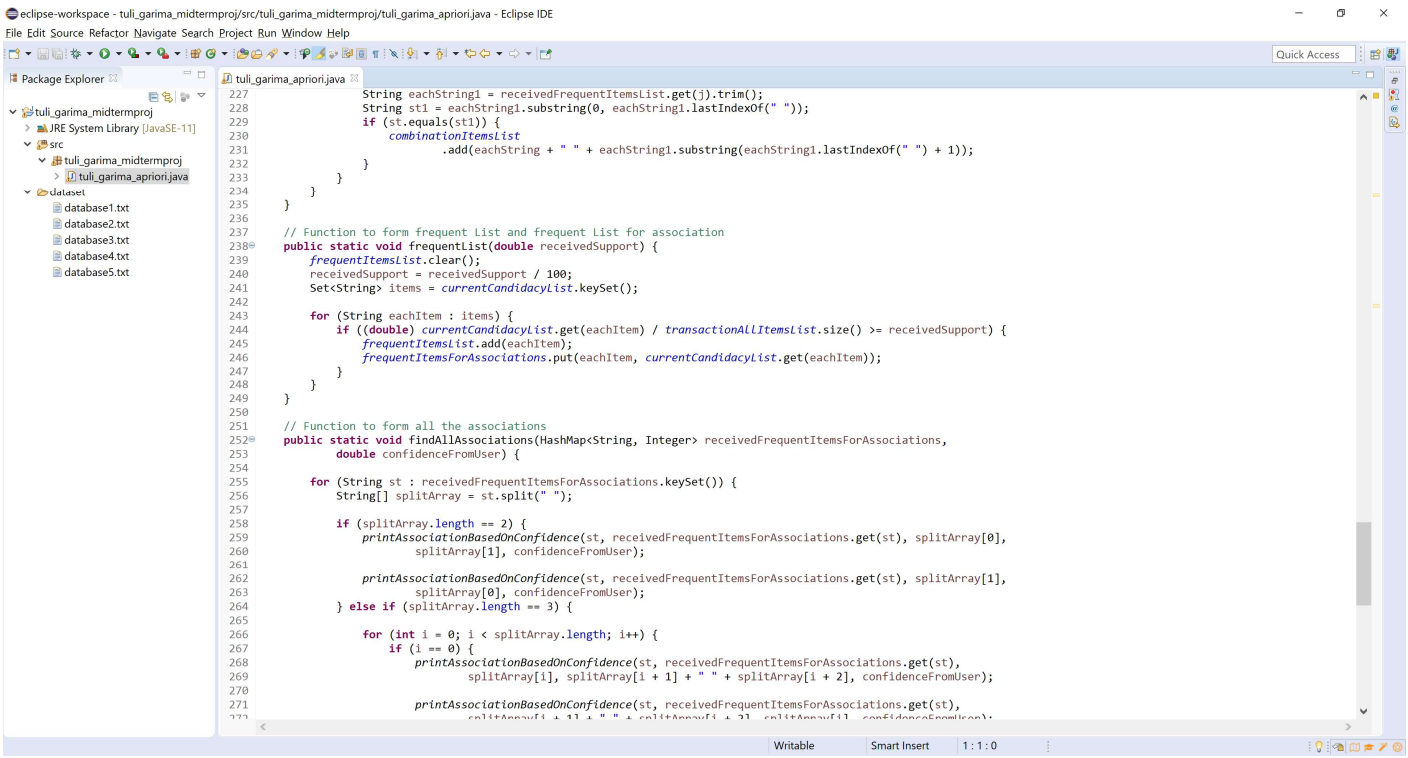


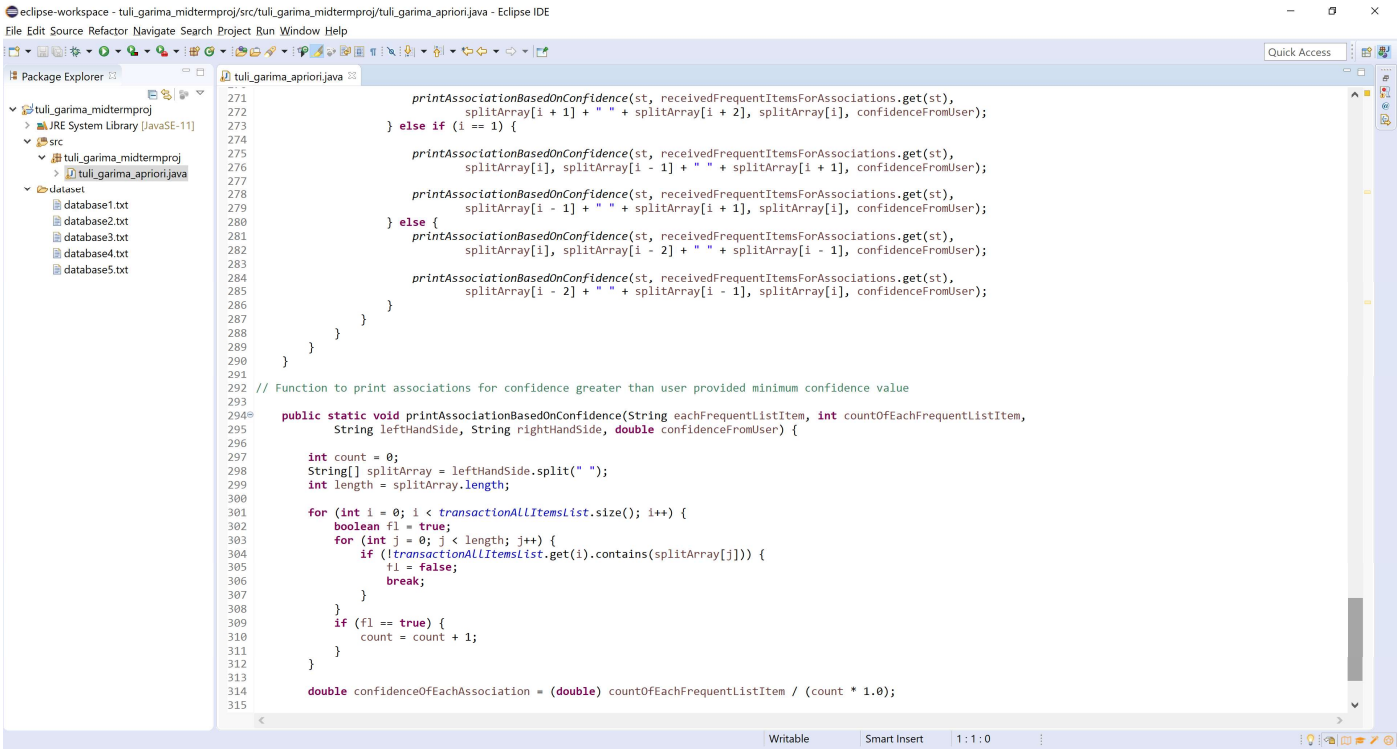


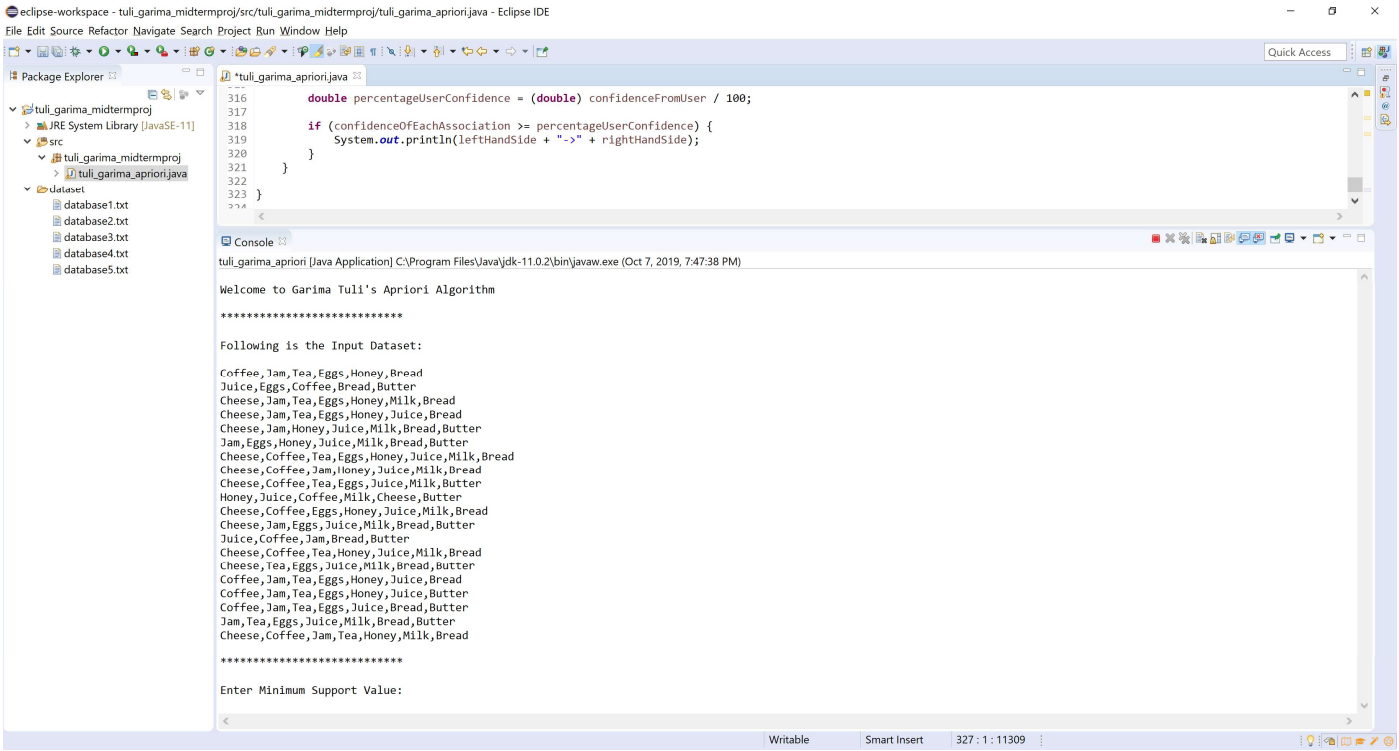

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
findAllAssociations(frequentItemsForAssociations, confidence);
}
// Read Data from File
public static void readDataFromFile() {
// Replace different file name here for different data set
String fileName = "dataset\\database1.txt";
// Local variable for each line
String eachLine = null;
// Unique Items
HashSet<String> uniqueItems = new HashSet<String>();
try {
FileReader fileReader = new FileReader(fileName);
BufferedReader bufferedReader = new BufferedReader(fileReader);
while ((eachLine = bufferedReader.readLine()) != null) {
// Local Items to form transactionItemsList
HashSet<String> localItems = new HashSet<String>();
String formedList[] = eachLine.split(",");
for (int i = 0; i < formedList.length; i++) {
localItems.add(formedList[i]);
uniqueItems.add(formedList[i]);
}
System.out.println(eachLine);
// transactionItemsList will contain set of all itemset
transactionAllItemsList.add(localItems);
}
} catch (Exception e) {
System.out.println("In to Exception" + e);
}
// finalList is arraylist of unique items from itemset
uniqueItemsList = new ArrayList<String>(uniqueItems);
}
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
// Function to form candidacy list (for one or all possible combinations in all
// transactions)
public static void candidacyListforOneorAllCombinations(ArrayList<String> receivedUniqueItemsList,
int receivedCount) {
if (receivedCount == 1) {
currentCandidacyList.clear();
// Loop through final List which is a unique list (finalList)
// and all items (transactionItemsList)
// to generate count per item in each transaction
for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
String oneItem = receivedUniqueItemsList.get(i);
for (int j = 0; j < transactionAllItemsList.size(); j++) {
if (transactionAllItemsList.get(j).contains(oneItem)) {
if (currentCandidacyList.containsKey(oneItem)) {
int counter = currentCandidacyList.get(oneItem);
currentCandidacyList.put(oneItem, ++counter);
} else {
currentCandidacyList.put(oneItem, 1);
}
}
}
}
} else {
currentCandidacyList.clear();
for (int i = 0; i < receivedUniqueItemsList.size(); i++) {
String st = receivedUniqueItemsList.get(i);
String[] splitString = st.split(" ");
for (int j = 0; j < transactionAllItemsList.size(); j++) {
boolean fl = true;
for (int k = 0; k < splitString.length; k++) {
if (!transactionAllItemsList.get(j).contains(splitString[k])) {
fl = false;
break;
}
}
if (fl == true) {
if (currentCandidacyList.containsKey(st)) {
int value = currentCandidacyList.get(st);
currentCandidacyList.put(st, ++value);
} else {
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
currentCandidacyList.put(st, 1);
}
}
}
}
}
// Function to form ArrayList for two items per transaction
public static void ListWithTwoItems(ArrayList<String> receivedFrequentItemsList, int receivedFrequentItemsListSize,
int itemCountSize, int i, String[] itemCountArray, int j) {
if (i == itemCountSize) {
String st = "";
for (int k = 0; k < itemCountSize; k++) {
st = st + itemCountArray[k] + " ";
}
combinationItemsList.add(st);
return;
}
if (j >= receivedFrequentItemsListSize) {
return;
}
itemCountArray[i] = receivedFrequentItemsList.get(j);
ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize, itemCountSize, i + 1, itemCountArray,
j + 1);
ListWithTwoItems(receivedFrequentItemsList, receivedFrequentItemsListSize, itemCountSize, i, itemCountArray,
j + 1);
}
// Function to form ArrayList for three or more items per transaction
private static void ListWithThreeOrMoreItems(ArrayList<String> receivedFrequentItemsList, int receivedCount) {
combinationItemsList.clear();
for (int i = 0; i < receivedFrequentItemsList.size(); i++) {
String eachString = receivedFrequentItemsList.get(i).trim();
String st = eachString.substring(0, eachString.lastIndexOf(" "));
for (int j = i + 1; j < receivedFrequentItemsList.size(); j++) {
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
String eachString1 = receivedFrequentItemsList.get(j).trim();
String st1 = eachString1.substring(0, eachString1.lastIndexOf(" "));
if (st.equals(st1)) {
combinationItemsList
.add(eachString + " " + eachString1.substring(eachString1.lastIndexOf(" ") + 1));
}
}
}
}
// Function to form frequent List and frequent List for association
public static void frequentList(double receivedSupport) {
frequentItemsList.clear();
receivedSupport = receivedSupport / 100;
Set<String> items = currentCandidacyList.keySet();
for (String eachItem : items) {
if ((double) currentCandidacyList.get(eachItem) / transactionAllItemsList.size() >= receivedSupport) {
frequentItemsList.add(eachItem);
frequentItemsForAssociations.put(eachItem, currentCandidacyList.get(eachItem));
}
}
}
// Function to form all the associations
public static void findAllAssociations(HashMap<String, Integer> receivedFrequentItemsForAssociations,
double confidenceFromUser) {
for (String st : receivedFrequentItemsForAssociations.keySet()) {
String[] splitArray = st.split(" ");
if (splitArray.length == 2) {
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[0],
splitArray[1], confidenceFromUser);
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st), splitArray[1],
splitArray[0], confidenceFromUser);
} else if (splitArray.length == 3) {
for (int i = 0; i < splitArray.length; i++) {
if (i == 0) {
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i], splitArray[i + 1] + " " + splitArray[i + 2], confidenceFromUser);
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i + 1] + " " + splitArray[i + 2], splitArray[i], confidenceFromUser);
} else if (i == 1) {
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i], splitArray[i - 1] + " " + splitArray[i + 1], confidenceFromUser);
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i - 1] + " " + splitArray[i + 1], splitArray[i], confidenceFromUser);
} else {
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i], splitArray[i - 2] + " " + splitArray[i - 1], confidenceFromUser);
printAssociationBasedOnConfidence(st, receivedFrequentItemsForAssociations.get(st),
splitArray[i - 2] + " " + splitArray[i - 1], splitArray[i], confidenceFromUser);
}
}
}
}
// Function to print associations for confidence greater than user provided minimum confidence value
public static void printAssociationBasedOnConfidence(String eachFrequentListItem, int countOfEachFrequentListItem,
String leftHandSide, String rightHandSide, double confidenceFromUser) {
int count = 0;
String[] splitArray = leftHandSide.split(" ");
int length = splitArray.length;
for (int i = 0; i < transactionAllItemsList.size(); i++) {
boolean fl = true;
for (int j = 0; j < length; j++) {
if (!transactionAllItemsList.get(i).contains(splitArray[j])) {
fl = false;
break;
}
}
if (fl == true) {
count = count + 1;
}
}
double confidenceOfEachAssociation = (double) countOfEachFrequentListItem / (count * 1.0);
Writable
Smart Insert
1 : 1 : 0

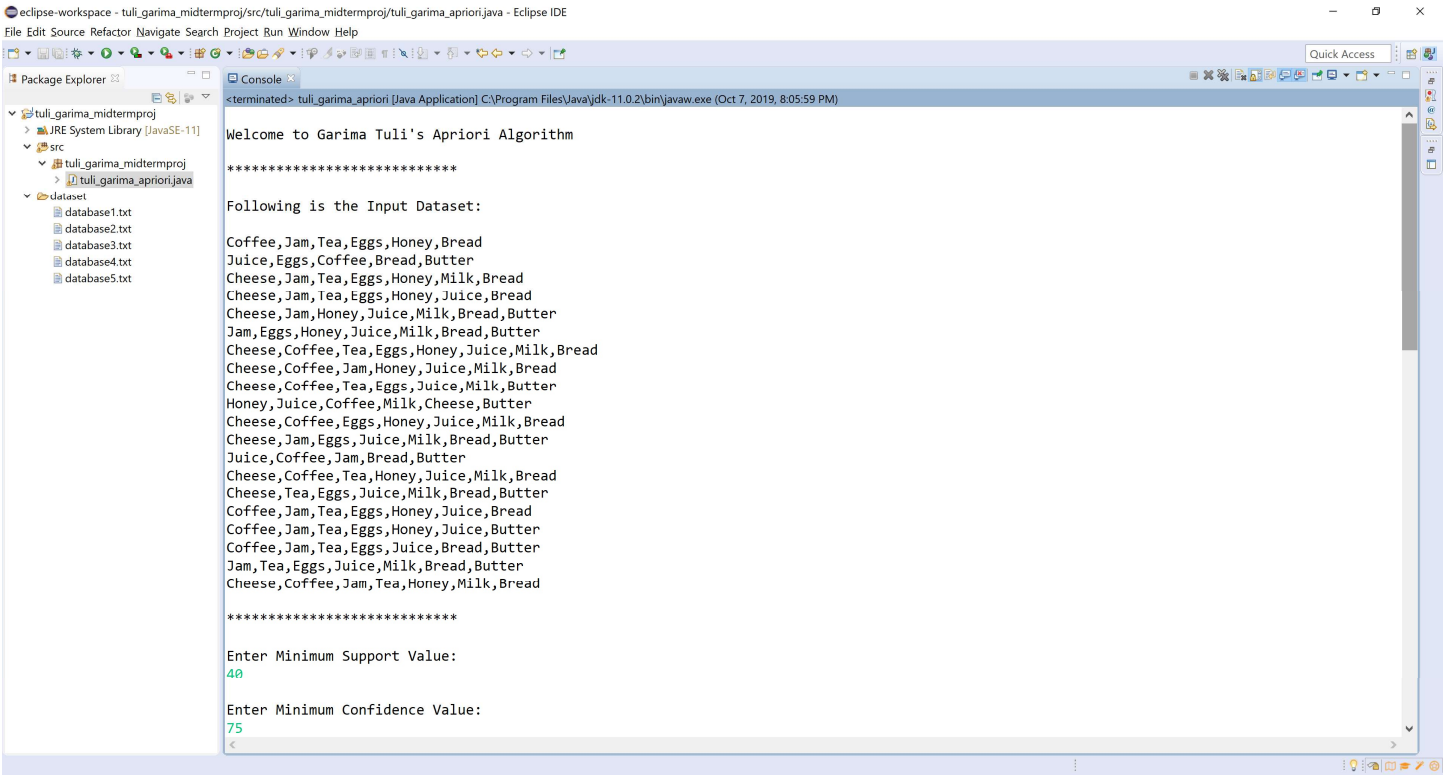

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
*tuli_garima_apriori.java
double percentageUserConfidence = (double) confidenceFromUser / 100;
if (confidenceOfEachAssociation >= percentageUserConfidence) {
System.out.println(leftHandSide + "->" + rightHandSide);
}
}
}
Console
tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 7:47:38 PM)
Welcome to Garima Tuli's Apriori Algorithm
****
Following is the Input Dataset:
Coffee,Jam,Tea,Eggs,Honey,Bread
Juice,Eggs,Coffee,Bread,Butter
Cheese,Jam,Tea,Eggs,Honey,Milk,Bread
Cheese,Jam,Tea,Eggs,Honey,Juice,Bread
Cheese,Jam,Honey,Juice,Milk,Bread,Butter
Jam,Eggs,Honey,Juice,Milk,Bread,Butter
Cheese,Coffee,Tea,Eggs,Honey,Juice,Milk,Bread
Cheese,Coffee,Jam,Honey,Juice,Milk,Bread
Cheese,Coffee,Tea,Eggs,Juice,Milk,Butter
Honey,Juice,Coffee,Milk,Cheese,Butter
Cheese,Coffee,Eggs,Honey,Juice,Milk,Bread
Cheese,Jam,Eggs,Juice,Milk,Bread,Butter
Juice,Coffee,Jam,Bread,Butter
Cheese,Coffee,Tea,Honey,Juice,Milk,Bread
Cheese,Tea,Eggs,Juice,Milk,Bread,Butter
Coffee,Jam,Tea,Eggs,Honey,Juice,Bread
Coffee,Jam,Tea,Eggs,Honey,Juice,Butter
Coffee,Jam,Tea,Eggs,Juice,Bread,Butter
Jam,Tea,Eggs,Juice,Milk,Bread,Butter
Cheese,Coffee,Jam,Tea,Honey,Milk,Bread
****
Enter Minimum Support Value:
Writable
Smart Insert
327 : 1 : 11309

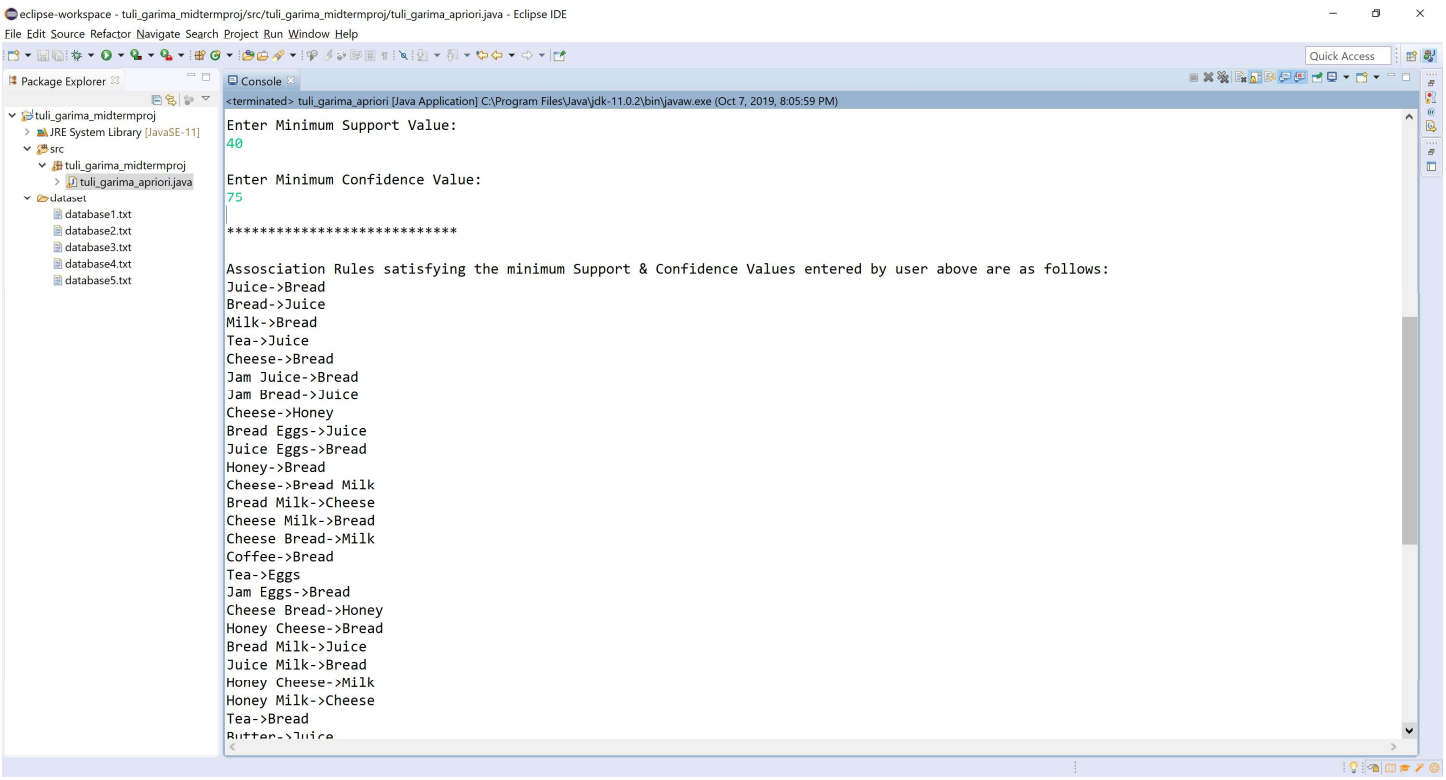
Output (Screenshots with user specified values for minimum support and confidence on 5 different datasets)

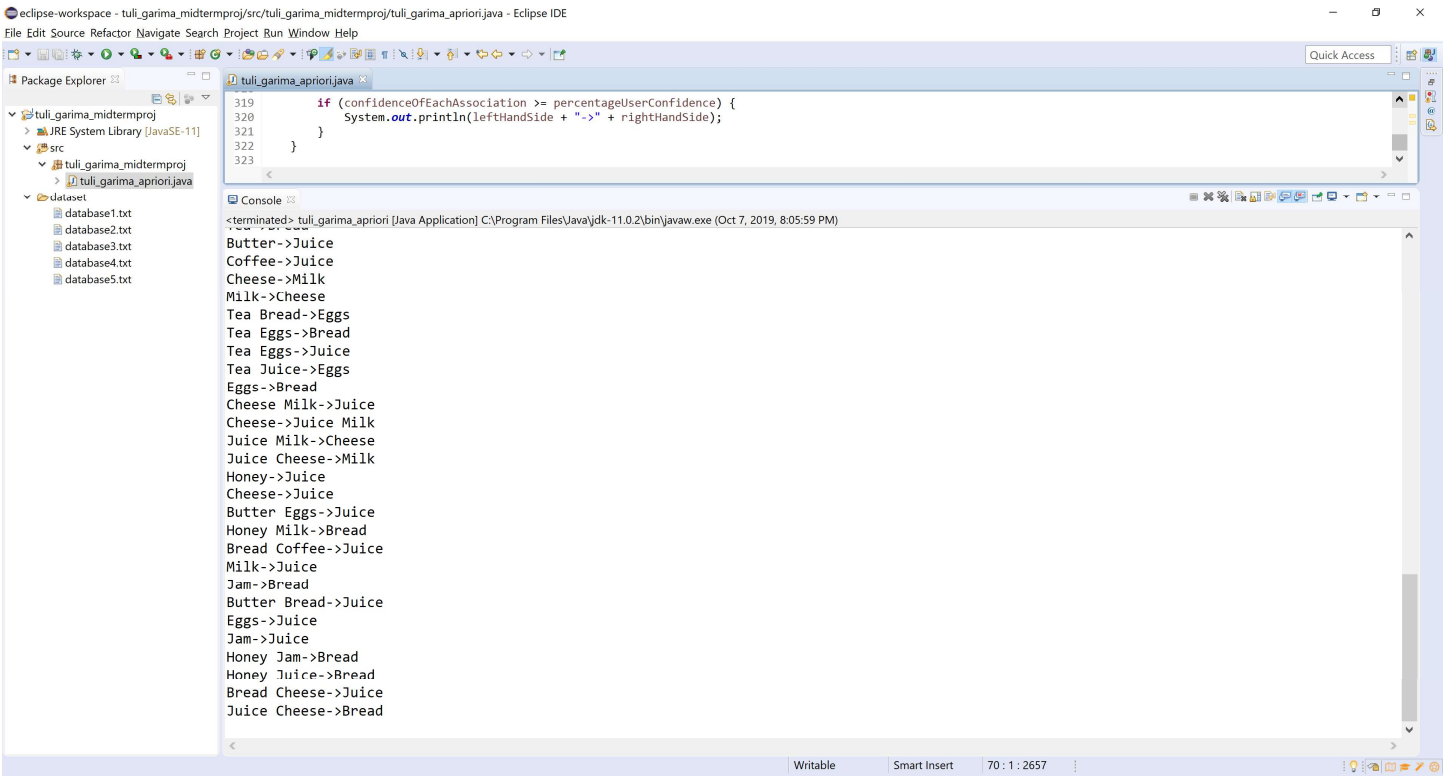
Screenshot of Output for Dataset 1:

User enters minimum support value as 40 and minimum confidence value as 75 (as displayed in green color in the screenshot below).

This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

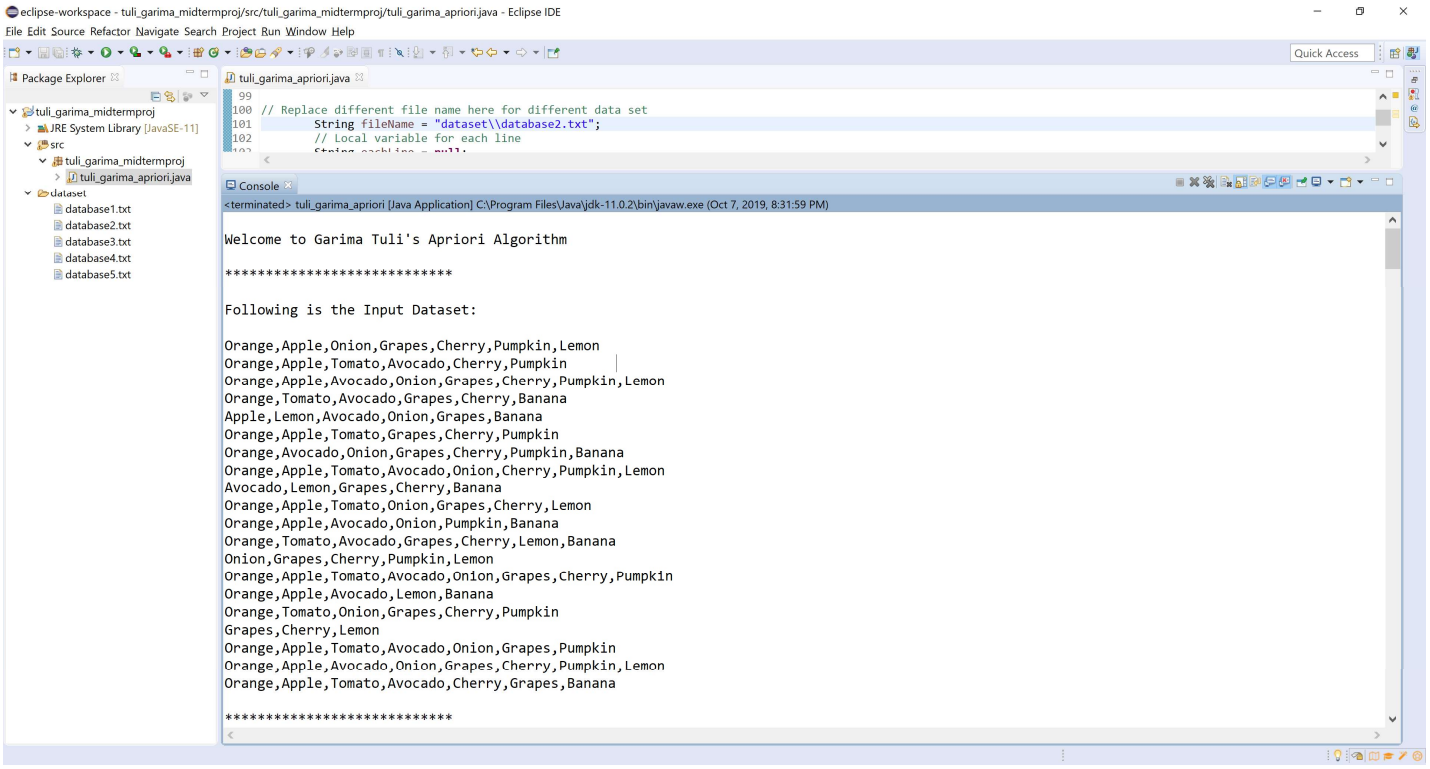


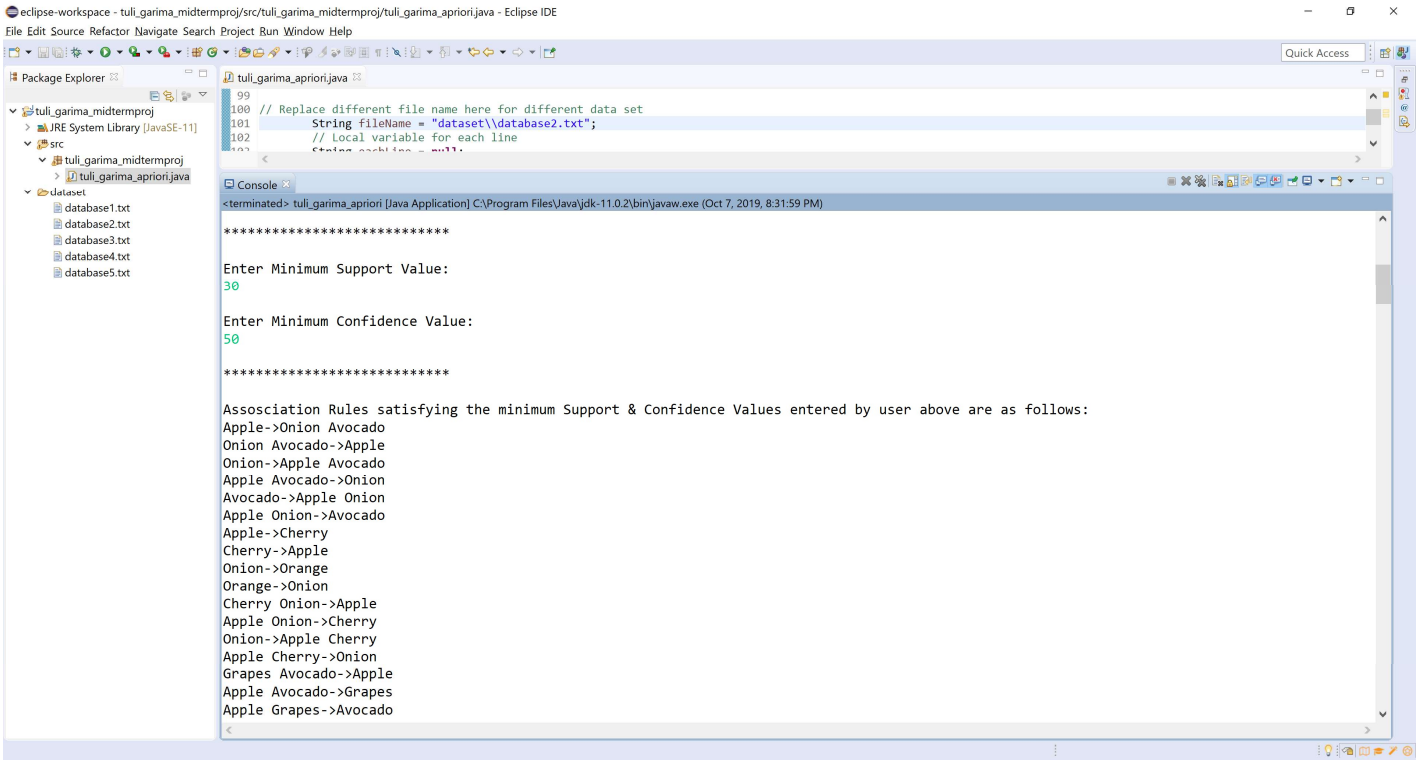

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:05:59 PM)
Enter Minimum Support Value:
40
Enter Minimum Confidence Value:
75
****
Association Rules satisfying the minimum Support & Confidence Values entered by user above are as follows:
Juice->Bread
Bread->Juice
Milk->Bread
Tea->Juice
Cheese->Bread
Jam Juice->Bread
Jam Bread->Juice
Cheese->Honey
Bread Eggs->Juice
Juice Eggs->Bread
Honey->Bread
Cheese->Bread Milk
Bread Milk->Cheese
Cheese Milk->Bread
Cheese Bread->Milk
Coffee->Bread
Tea->Eggs
Jam Eggs->Bread
Cheese Bread->Honey
Honey Cheese->Bread
Bread Milk->Juice
Juice Milk->Bread
Honey Cheese->Milk
Honey Milk->Cheese
Tea->Bread

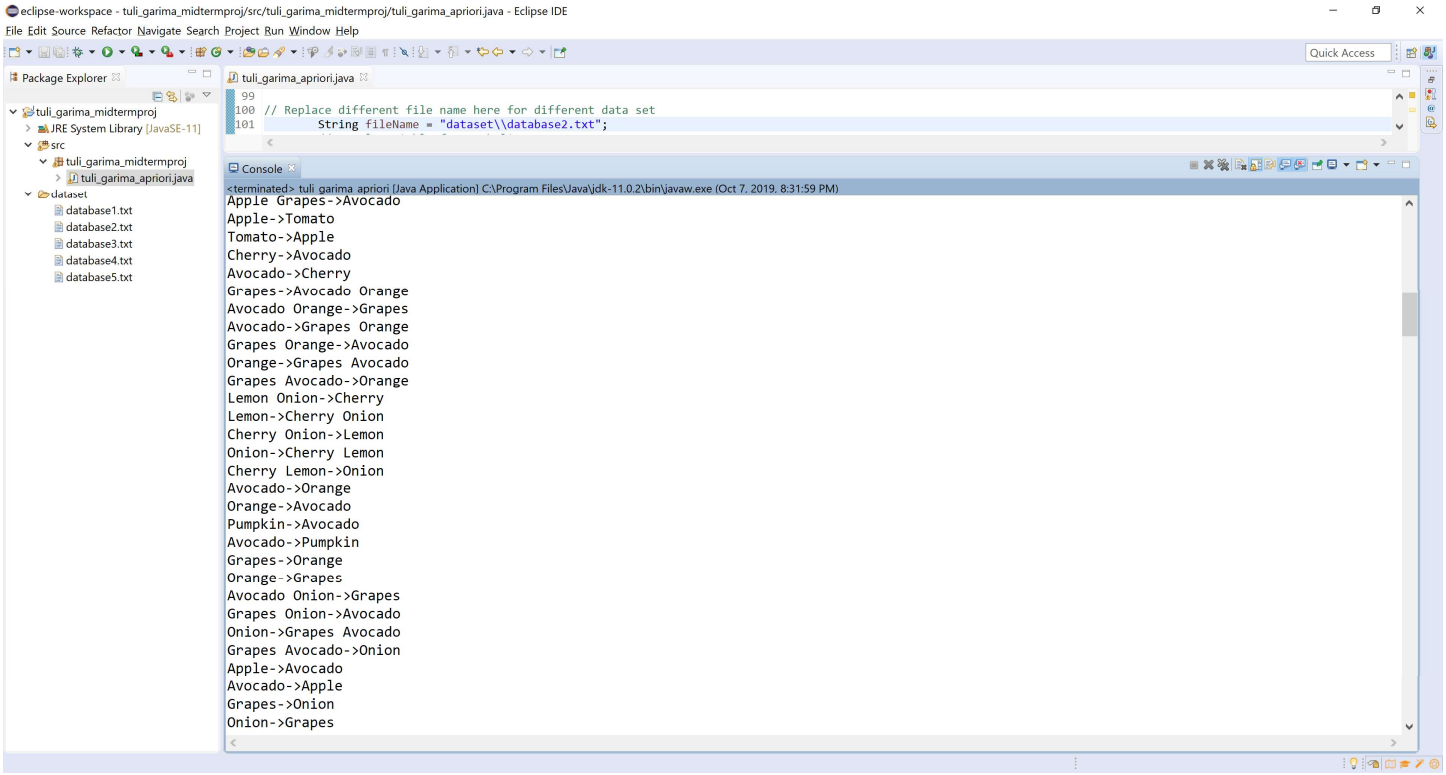

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
tuli_garima_apriori.java
if (confidenceOfEachAssociation >= percentageUserConfidence) {
System.out.println(leftHandSide + "->" + rightHandSide);
}
}
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:05:59 PM)
Butter->Juice
Coffee->Juice
Cheese->Milk
Milk->Cheese
Tea Bread->Eggs
Tea Eggs->Bread
Tea Eggs->Juice
Tea Juice->Eggs
Eggs->Bread
Cheese Milk->Juice
Cheese->Juice Milk
Juice Milk->Cheese
Juice Cheese->Milk
Honey->Juice
Cheese->Juice
Butter Eggs->Juice
Honey Milk->Bread
Bread Coffee->Juice
Milk->Juice
Jam->Bread
Butter Bread->Juice
Eggs->Juice
Jam->Juice
Honey Jam->Bread
Bread Cheese->Juice
Juice Cheese->Bread
Writable
Smart Insert
70 : 1 : 2657

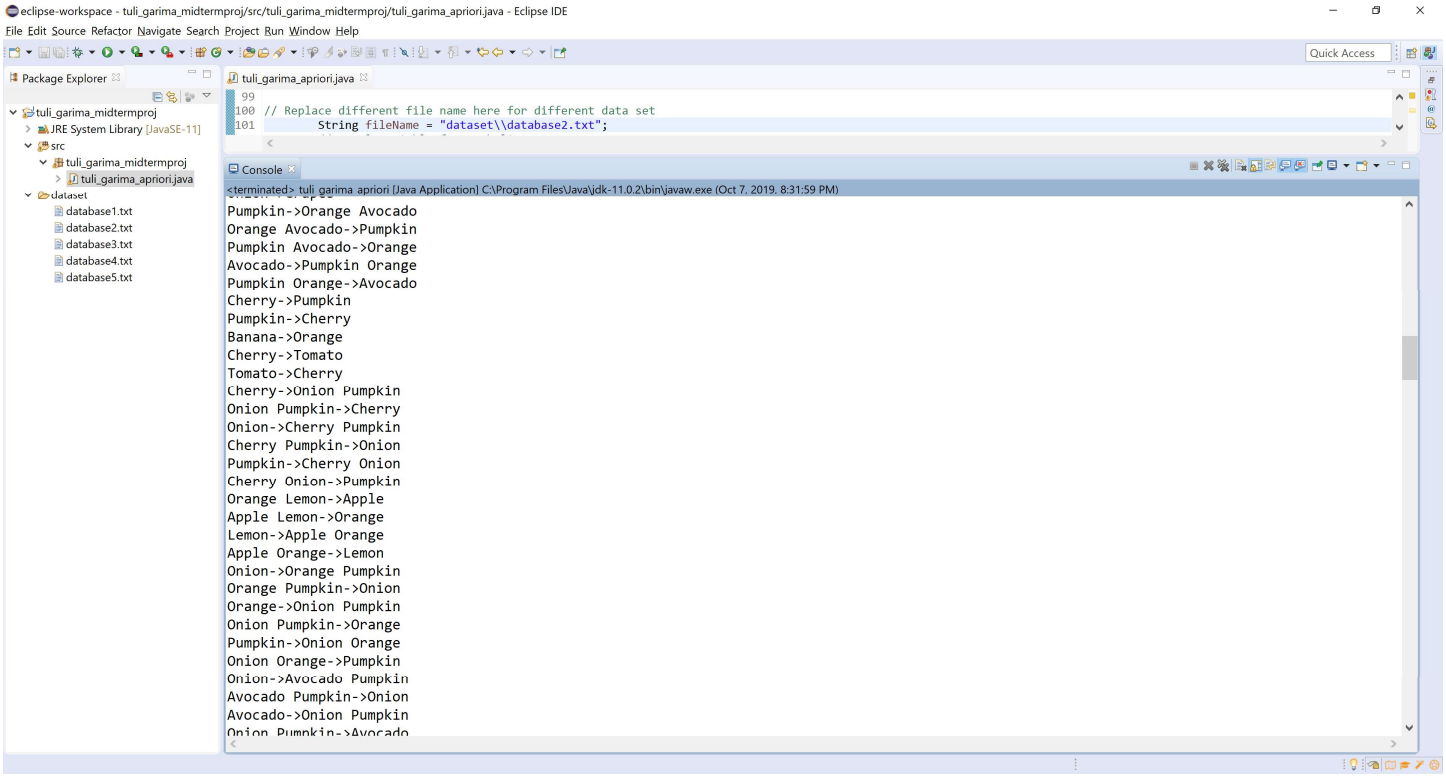
Screenshot of Output for Dataset 2:

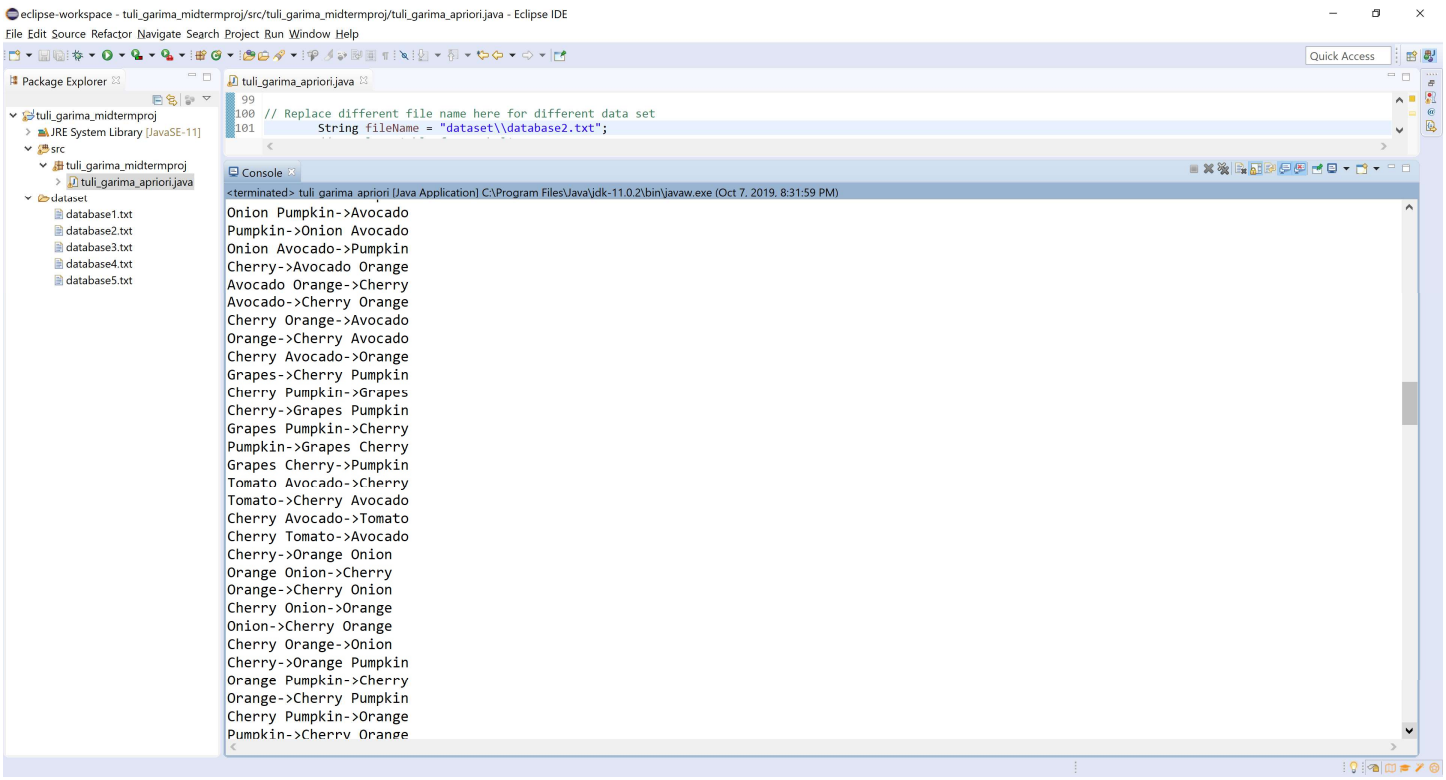
User enters minimum support value as 30 and minimum confidence value as 50 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

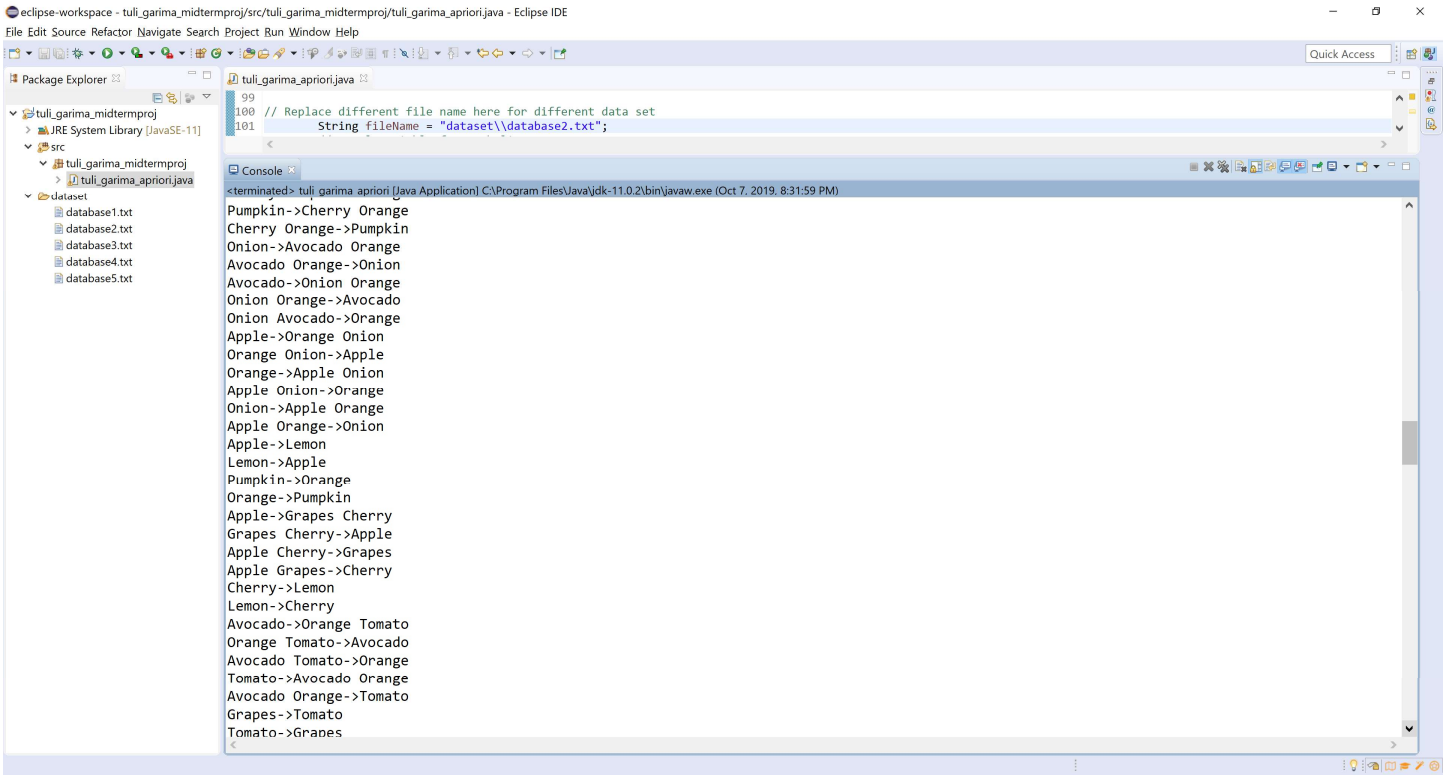


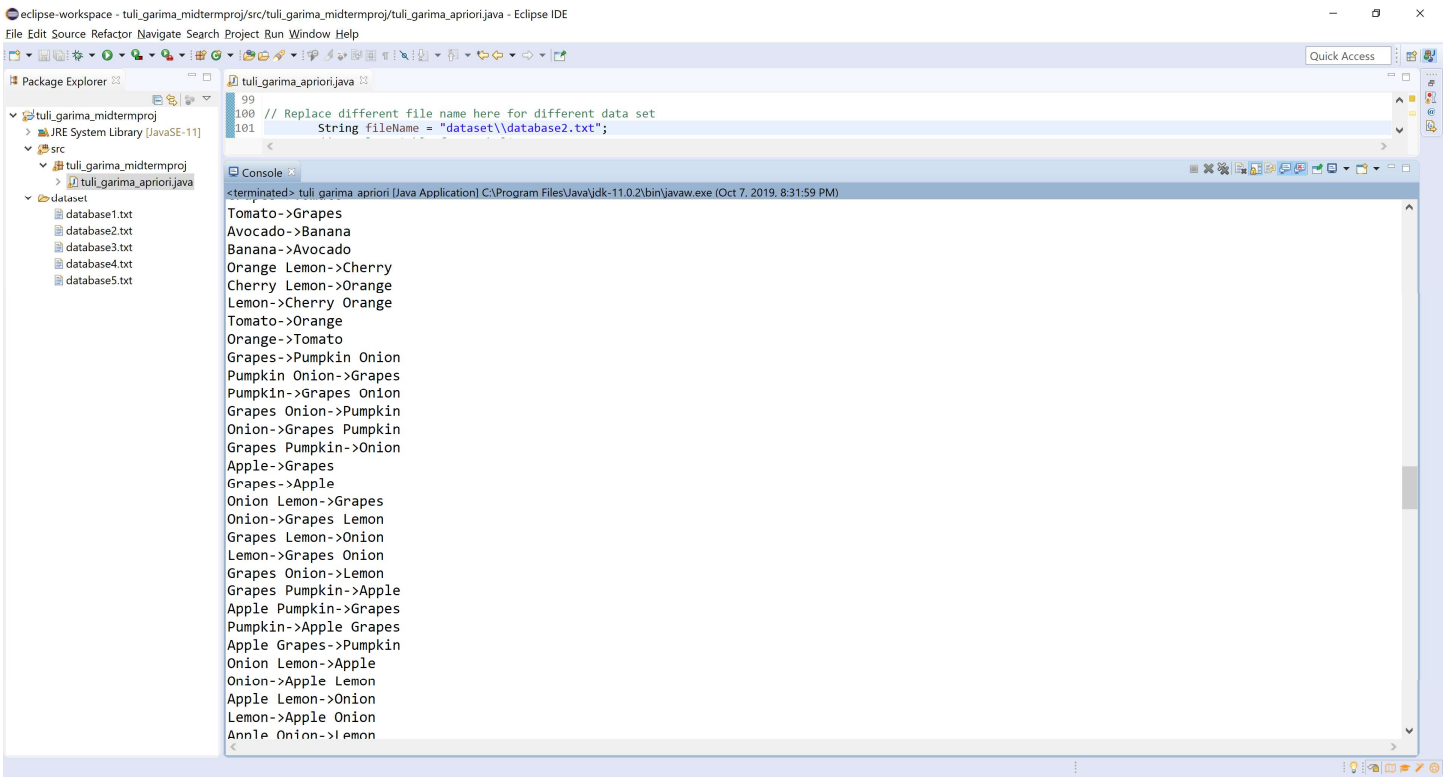


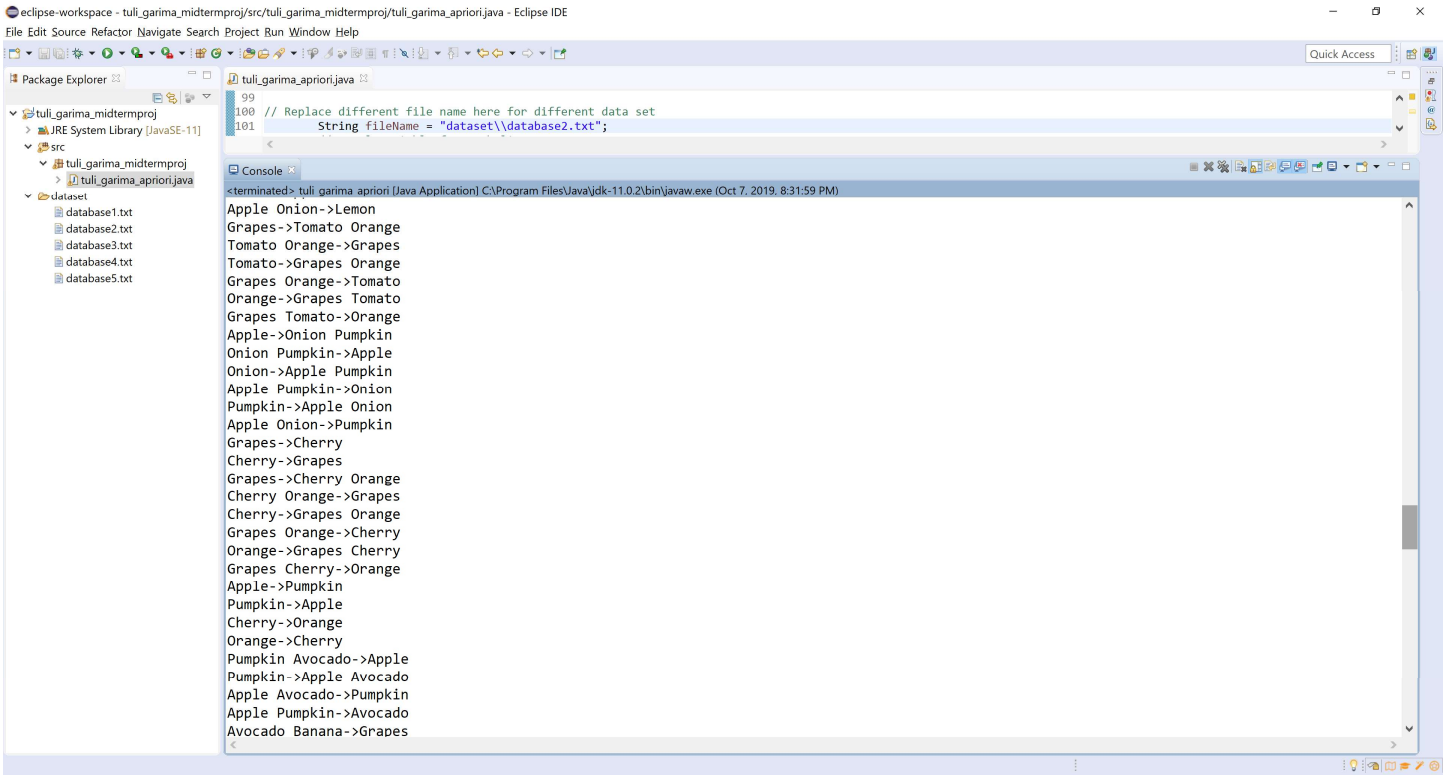

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Apple Grapes->Avocado
Apple->Tomato
Tomato->Apple
Cherry->Avocado
Avocado->Cherry
Grapes->Avocado Orange
Avocado Orange->Grapes
Avocado->Grapes Orange
Grapes Orange->Avocado
Orange->Grapes Avocado
Grapes Avocado->Orange
Lemon Onion->Cherry
Lemon->Cherry Onion
Cherry Onion->Lemon
Onion->Cherry Lemon
Cherry Lemon->Onion
Avocado->Orange
Orange->Avocado
Pumpkin->Avocado
Avocado->Pumpkin
Grapes->Orange
Orange->Grapes
Avocado Onion->Grapes
Grapes Onion->Avocado
Onion->Grapes Avocado
Grapes Avocado->Onion
Apple->Avocado
Avocado->Apple
Grapes->Onion
Onion->Grapes

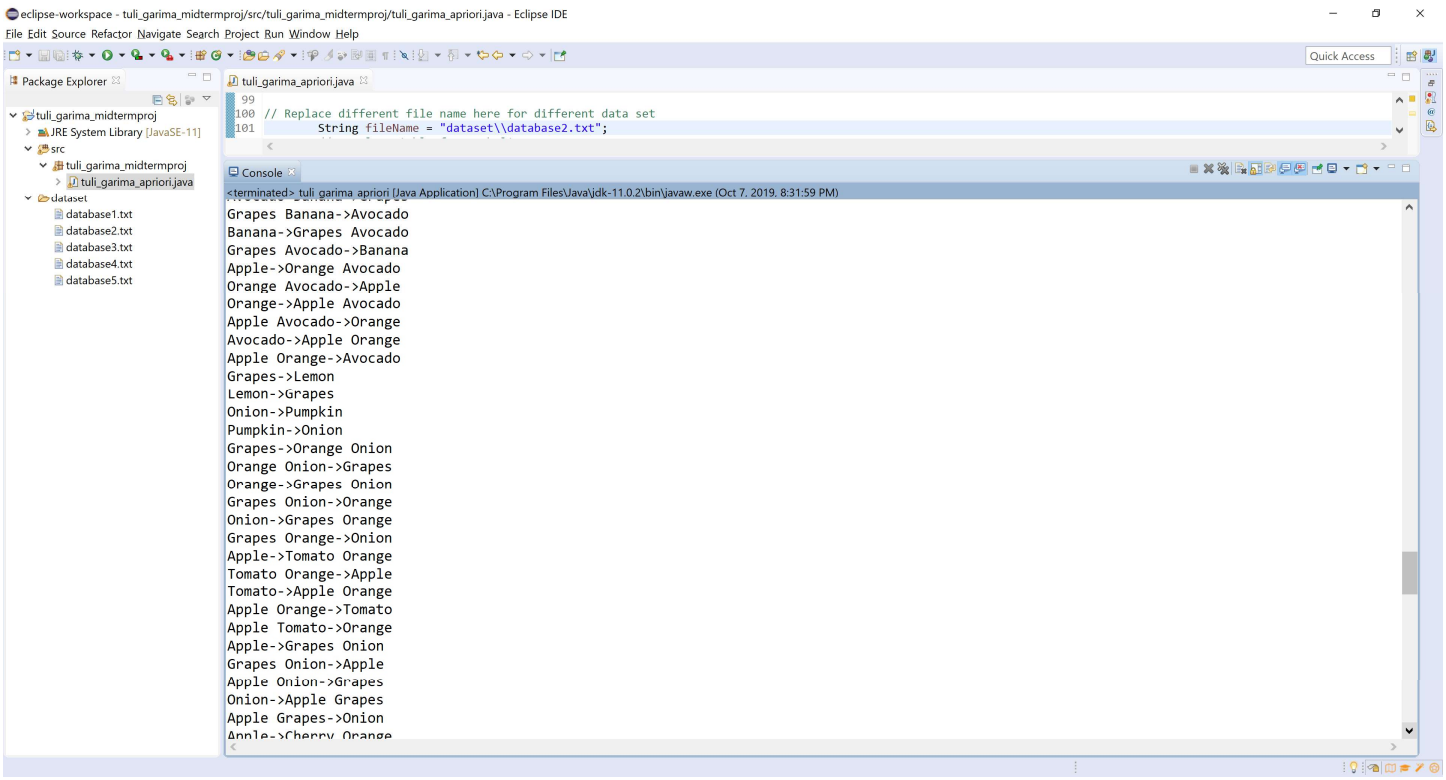

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Pumpkin->Orange Avocado
Orange Avocado->Pumpkin
Pumpkin Avocado->Orange
Avocado->Pumpkin Orange
Pumpkin Orange->Avocado
Cherry->Pumpkin
Pumpkin->Cherry
Banana->Orange
Cherry->Tomato
Tomato->Cherry
Cherry->Onion Pumpkin
Onion Pumpkin->Cherry
Onion->Cherry Pumpkin
Cherry Pumpkin->Onion
Pumpkin->Cherry Onion
Cherry Onion->Pumpkin
Orange Lemon->Apple
Apple Lemon->Orange
Lemon->Apple Orange
Apple Orange->Lemon
Onion->Orange Pumpkin
Orange Pumpkin->Onion
Orange->Onion Pumpkin
Onion Pumpkin->Orange
Pumpkin->Onion Orange
Onion Orange->Pumpkin
Onion->Avocado Pumpkin
Avocado Pumpkin->Onion
Avocado->Onion Pumpkin

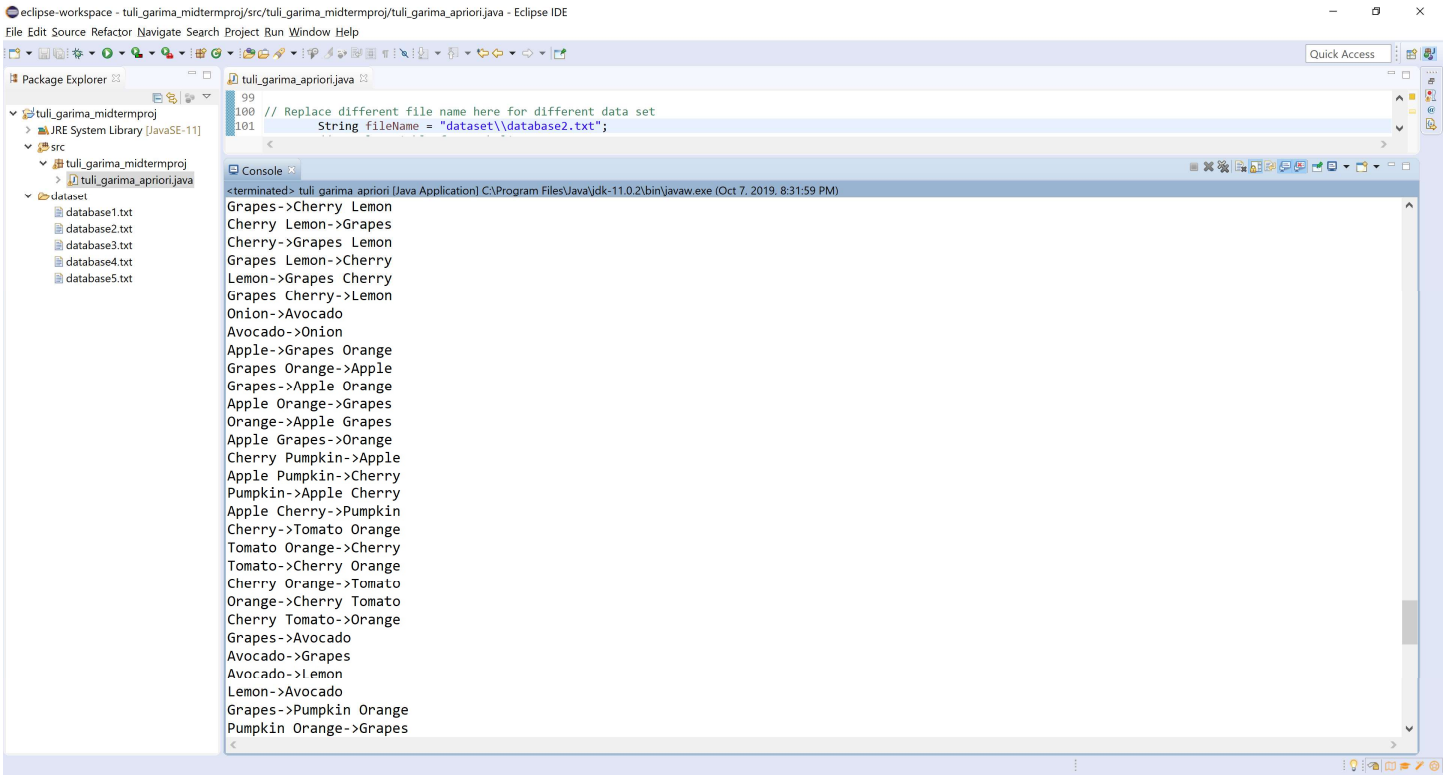

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Onion Pumpkin->Avocado
Pumpkin->Onion Avocado
Onion Avocado->Pumpkin
Cherry->Avocado Orange
Avocado Orange->Cherry
Avocado->Cherry Orange
Cherry Orange->Avocado
Orange->Cherry Avocado
Cherry Avocado->Orange
Grapes->Cherry Pumpkin
Cherry Pumpkin->Grapes
Cherry->Grapes Pumpkin
Grapes Pumpkin->Cherry
Pumpkin->Grapes Cherry
Grapes Cherry->Pumpkin
Tomato Avocado->Cherry
Tomato->Cherry Avocado
Cherry Avocado->Tomato
Cherry Tomato->Avocado
Cherry->Orange Onion
Orange Onion->Cherry
Orange->Cherry Onion
Cherry Onion->Orange
Onion->Cherry Orange
Cherry Orange->Onion
Cherry->Orange Pumpkin
Orange Pumpkin->Cherry
Orange->Cherry Pumpkin
Cherry Pumpkin->Orange
Pumpkin->Cherry Orange

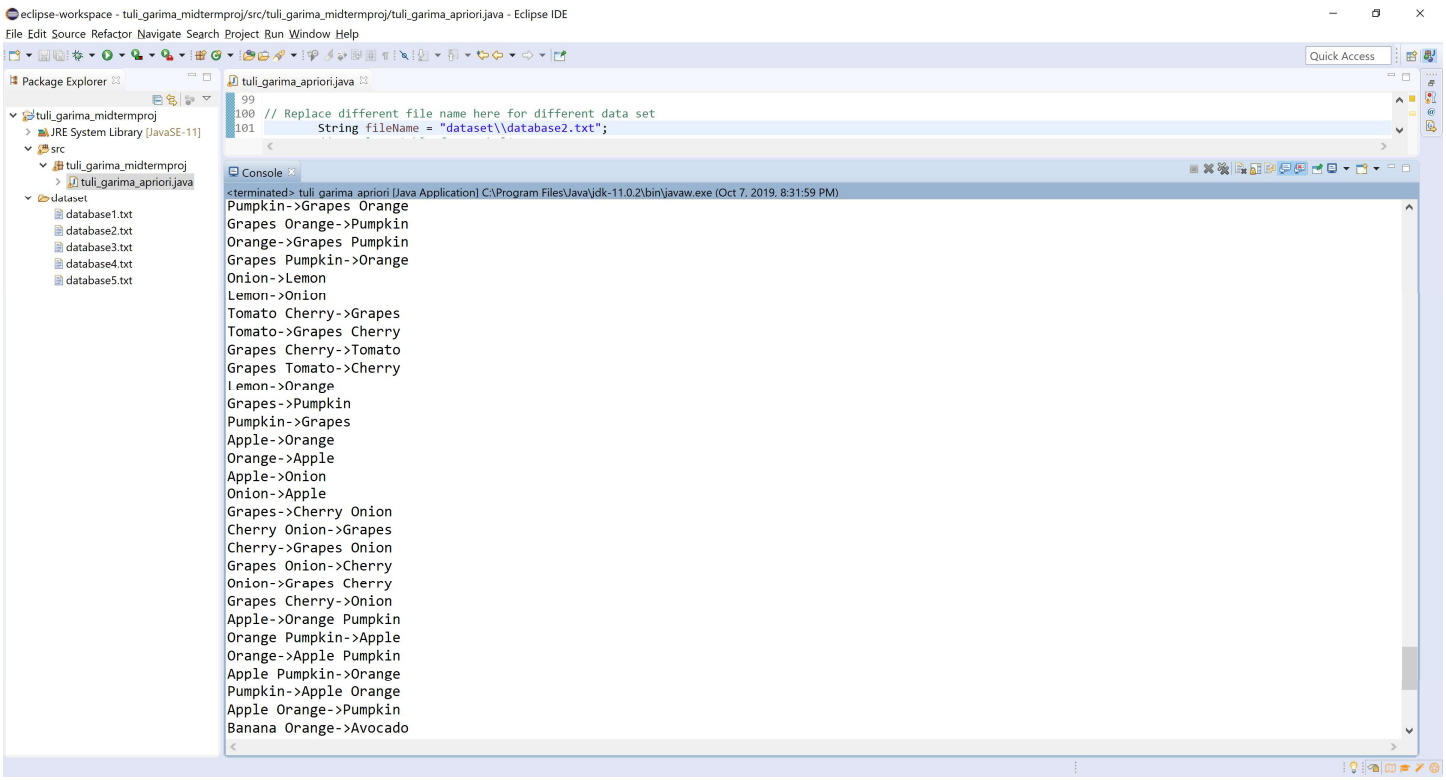

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Pumpkin->Cherry Orange
Cherry Orange->Pumpkin
Onion->Avocado Orange
Avocado Orange->Onion
Avocado->Onion Orange
Onion Orange->Avocado
Onion Avocado->Orange
Apple->Orange Onion
Orange Onion->Apple
Orange->Apple Onion
Apple Onion->Orange
Onion->Apple Orange
Apple Orange->Onion
Apple->Lemon
Lemon->Apple
Pumpkin->Orange
Orange->Pumpkin
Apple->Grapes Cherry
Grapes Cherry->Apple
Apple Cherry->Grapes
Apple Grapes->Cherry
Cherry->Lemon
Lemon->Cherry
Avocado->Orange Tomato
Orange Tomato->Avocado
Avocado Tomato->Orange
Tomato->Avocado Orange
Avocado Orange->Tomato
Grapes->Tomato
Tomato->Grapes

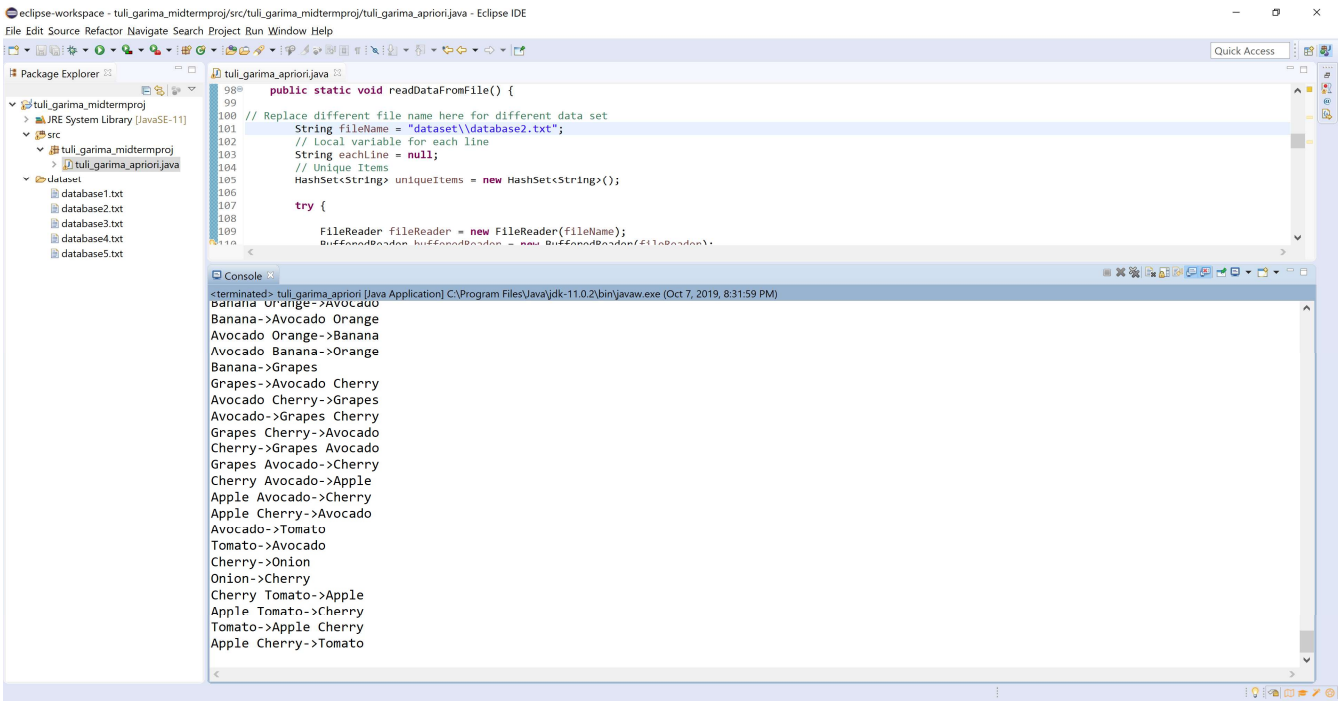

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Tomato->Grapes
Avocado->Banana
Banana->Avocado
Orange Lemon->Cherry
Cherry Lemon->Orange
Lemon->Cherry Orange
Tomato->Orange
Orange->Tomato
Grapes->Pumpkin Onion
Pumpkin Onion->Grapes
Pumpkin->Grapes Onion
Grapes Onion->Pumpkin
Onion->Grapes Pumpkin
Grapes Pumpkin->Onion
Apple->Grapes
Grapes->Apple
Onion Lemon->Grapes
Onion->Grapes Lemon
Grapes Lemon->Onion
Lemon->Grapes Onion
Grapes Onion->Lemon
Grapes Pumpkin->Apple
Apple Pumpkin->Grapes
Pumpkin->Apple Grapes
Apple Grapes->Pumpkin
Onion Lemon->Apple
Onion->Apple Lemon
Apple Lemon->Onion
Lemon->Apple Onion
Apple Onion->Lemon


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Apple Onion->Lemon
Grapes->Tomato Orange
Tomato Orange->Grapes
Tomato->Grapes Orange
Grapes Orange->Tomato
Orange->Grapes Tomato
Grapes Tomato->Orange
Apple->Onion Pumpkin
Onion Pumpkin->Apple
Onion->Apple Pumpkin
Apple Pumpkin->Onion
Pumpkin->Apple Onion
Apple Onion->Pumpkin
Grapes->Cherry
Cherry->Grapes
Grapes->Cherry Orange
Cherry Orange->Grapes
Cherry->Grapes Orange
Grapes Orange->Cherry
Orange->Grapes Cherry
Grapes Cherry->Orange
Apple->Pumpkin
Pumpkin->Apple
Cherry->Orange
Orange->Cherry
Pumpkin Avocado->Apple
Pumpkin->Apple Avocado
Apple Avocado->Pumpkin
Apple Pumpkin->Avocado
Avocado Banana->Grapes


eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Grapes Banana->Avocado
Banana->Grapes Avocado
Grapes Avocado->Banana
Apple->Orange Avocado
Orange Avocado->Apple
Orange->Apple Avocado
Apple Avocado->Orange
Avocado->Apple Orange
Apple Orange->Avocado
Grapes->Lemon
Lemon->Grapes
Onion->Pumpkin
Pumpkin->Onion
Grapes->Orange Onion
Orange Onion->Grapes
Orange->Grapes Onion
Grapes Onion->Orange
Onion->Grapes Orange
Grapes Orange->Onion
Apple->Tomato Orange
Tomato Orange->Apple
Tomato->Apple Orange
Apple Orange->Tomato
Apple Tomato->Orange
Apple->Grapes Onion
Grapes Onion->Apple
Apple Onion->Grapes
Onion->Apple Grapes
Apple Grapes->Onion

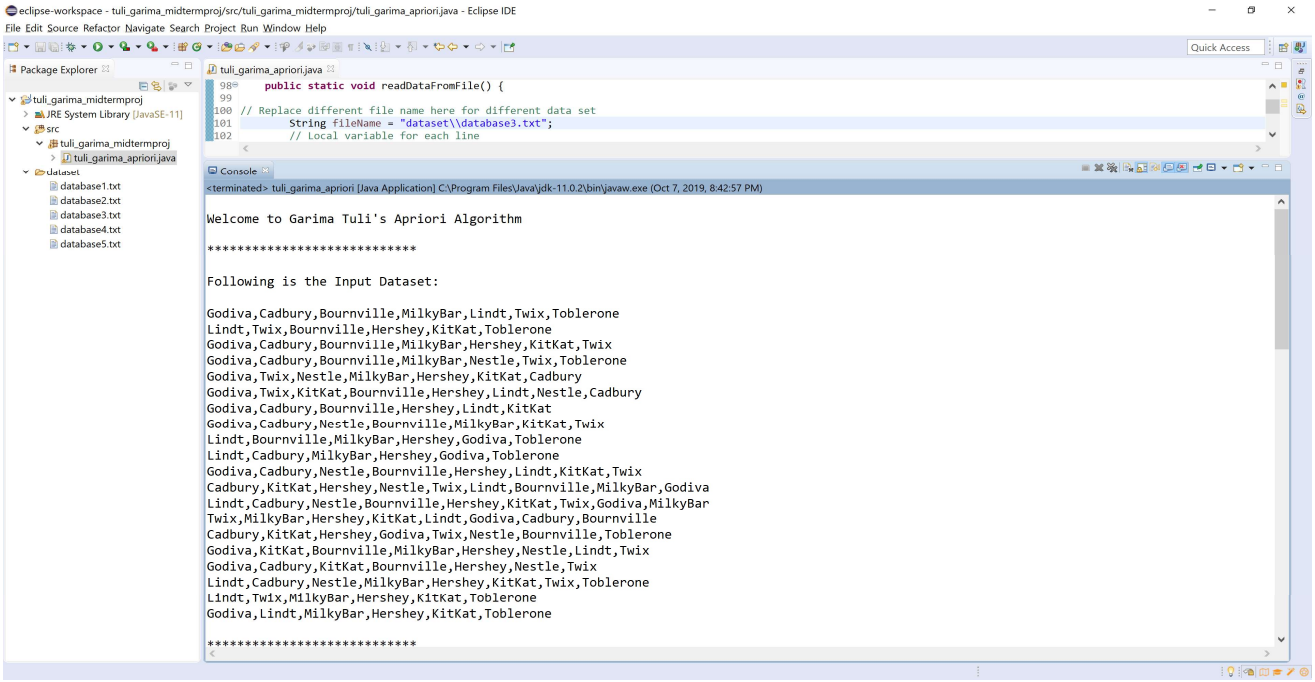

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Grapes->Cherry Lemon
Cherry Lemon->Grapes
Cherry->Grapes Lemon
Grapes Lemon->Cherry
Lemon->Grapes Cherry
Grapes Cherry->Lemon
Onion->Avocado
Avocado->Onion
Apple->Grapes Orange
Grapes Orange->Apple
Grapes->Apple Orange
Apple Orange->Grapes
Orange->Apple Grapes
Apple Grapes->Orange
Cherry Pumpkin->Apple
Apple Pumpkin->Cherry
Pumpkin->Apple Cherry
Apple Cherry->Pumpkin
Cherry->Tomato Orange
Tomato Orange->Cherry
Tomato->Cherry Orange
Cherry Orange->Tomato
Orange->Cherry Tomato
Cherry Tomato->Orange
Grapes->Avocado
Avocado->Grapes
Avocado->Lemon
Lemon->Avocado
Grapes->Pumpkin Orange
Pumpkin Orange->Grapes

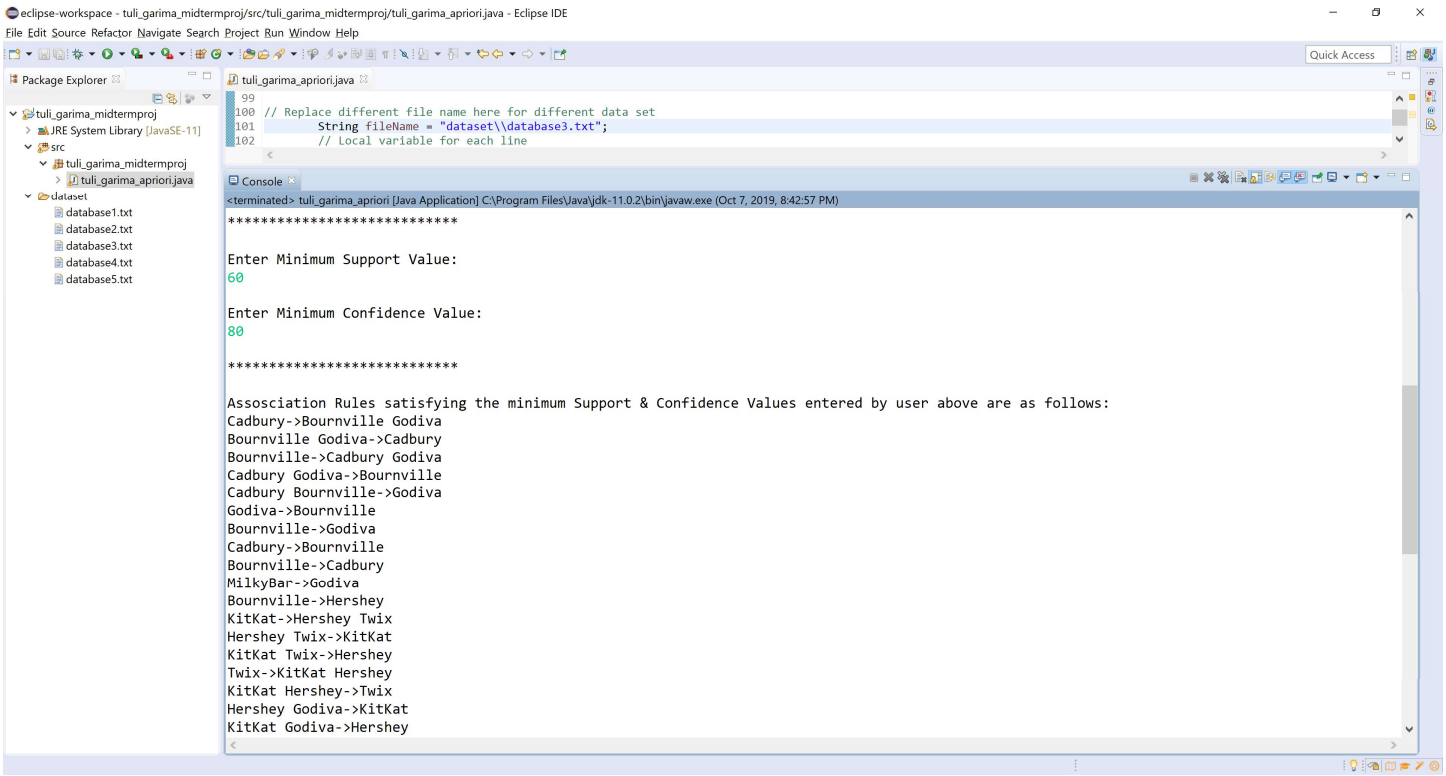

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Pumpkin->Grapes Orange
Grapes Orange->Pumpkin
Orange->Grapes Pumpkin
Grapes Pumpkin->Orange
Onion->Lemon
Lemon->Onion
Tomato Cherry->Grapes
Tomato->Grapes Cherry
Grapes Cherry->Tomato
Grapes Tomato->Cherry
Lemon->Orange
Grapes->Pumpkin
Pumpkin->Grapes
Apple->Orange
Orange->Apple
Apple->Onion
Onion->Apple
Grapes->Cherry Onion
Cherry Onion->Grapes
Cherry->Grapes Onion
Grapes Onion->Cherry
Onion->Grapes Cherry
Grapes Cherry->Onion
Apple->Orange Pumpkin
Orange Pumpkin->Apple
Orange->Apple Pumpkin
Apple Pumpkin->Orange
Pumpkin->Apple Orange
Apple Orange->Pumpkin
Banana Orange->Avocado

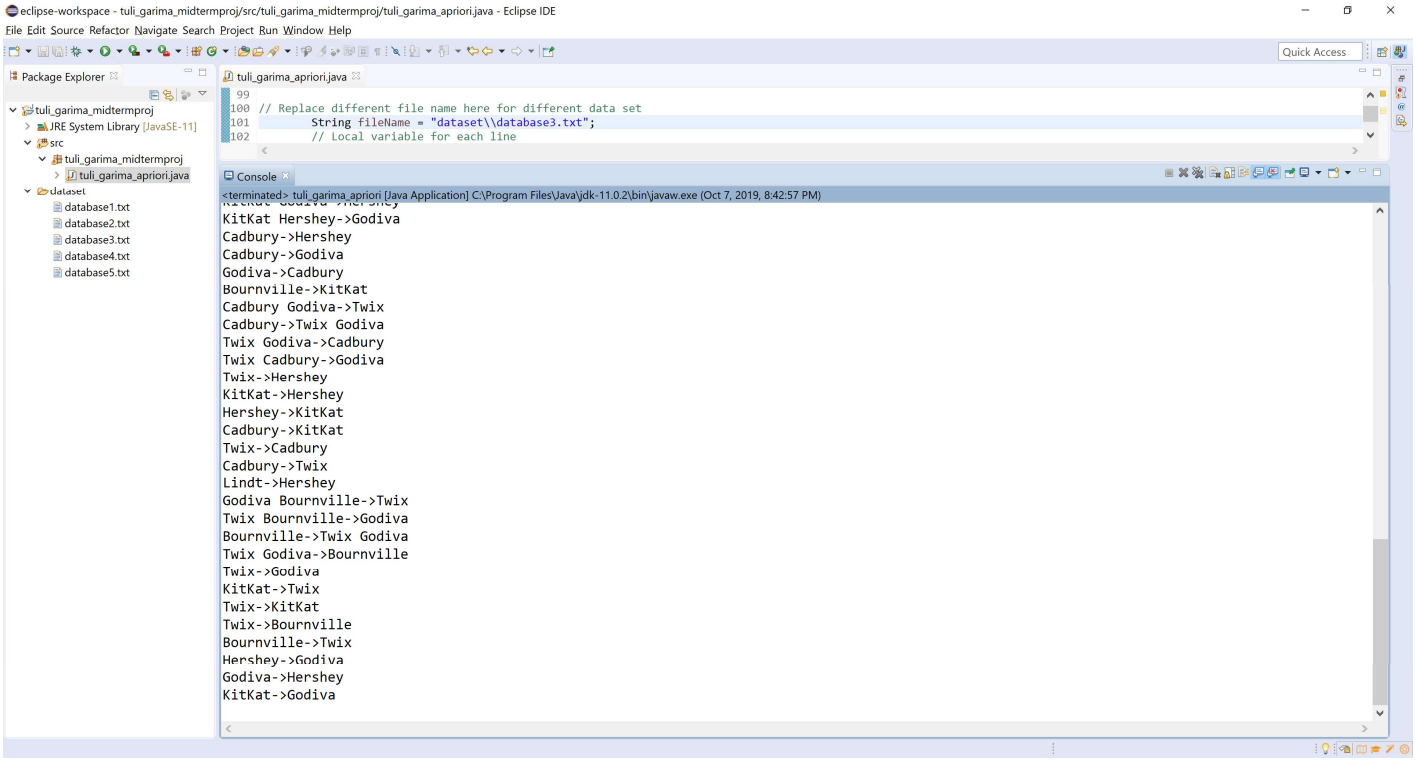

eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
98 public static void readDataFromFile() {
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database2.txt";
102 // Local variable for each line
103 String eachLine = null;
104 // Unique Items
105 HashSet<String> uniqueItems = new HashSet<String>();
106
107 try {
108
109 FileReader fileReader = new FileReader(fileName);
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:31:59 PM)
Banana->Avocado Orange
Avocado Orange->Banana
Avocado Banana->Orange
Banana->Grapes
Grapes->Avocado Cherry
Avocado Cherry->Grapes
Avocado->Grapes Cherry
Grapes Cherry->Avocado
Cherry->Grapes Avocado
Grapes Avocado->Cherry
Cherry Avocado->Apple
Apple Avocado->Cherry
Apple Cherry->Avocado
Avocado->Tomato
Tomato->Avocado
Cherry->Onion
Onion->Cherry
Cherry Tomato->Apple
Apple Tomato->Cherry
Tomato->Apple Cherry
Apple Cherry->Tomato

Screenshot of Output for Dataset 3:

User enters minimum support value as 60 and minimum confidence value as 80 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

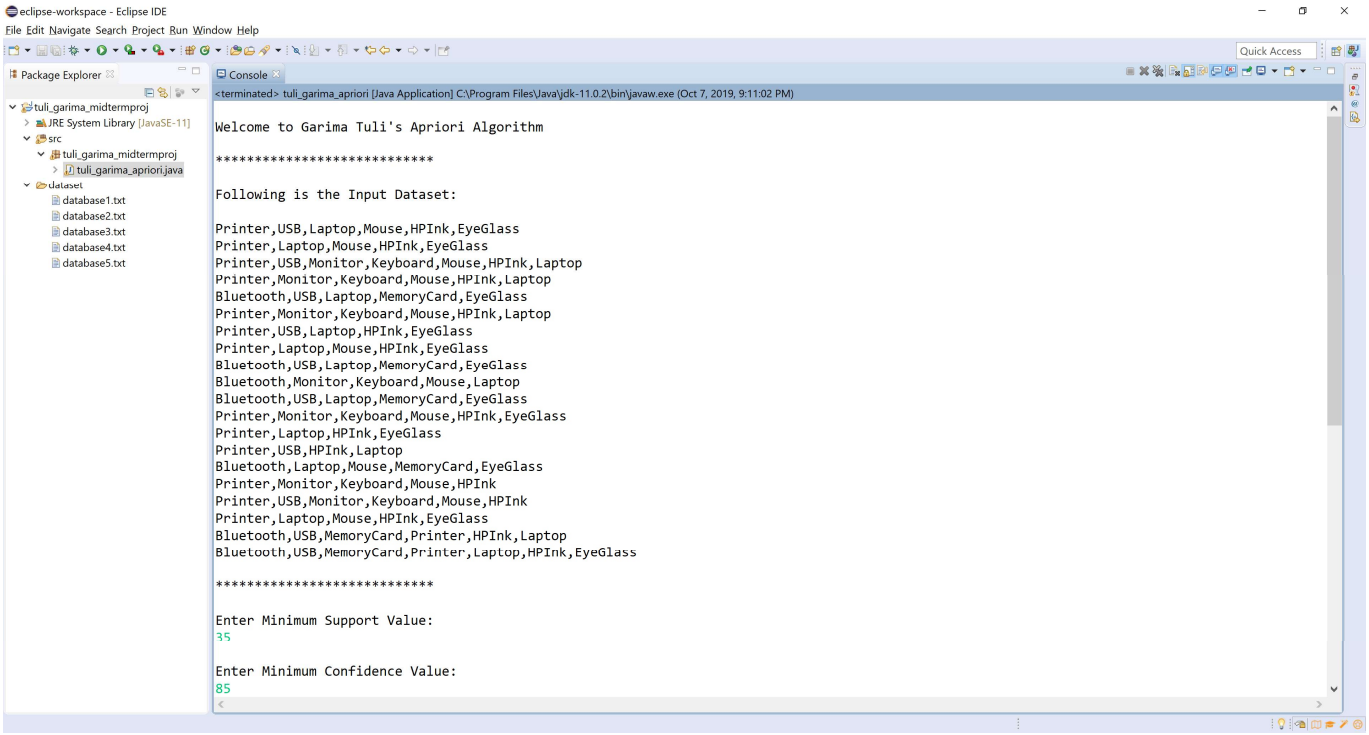


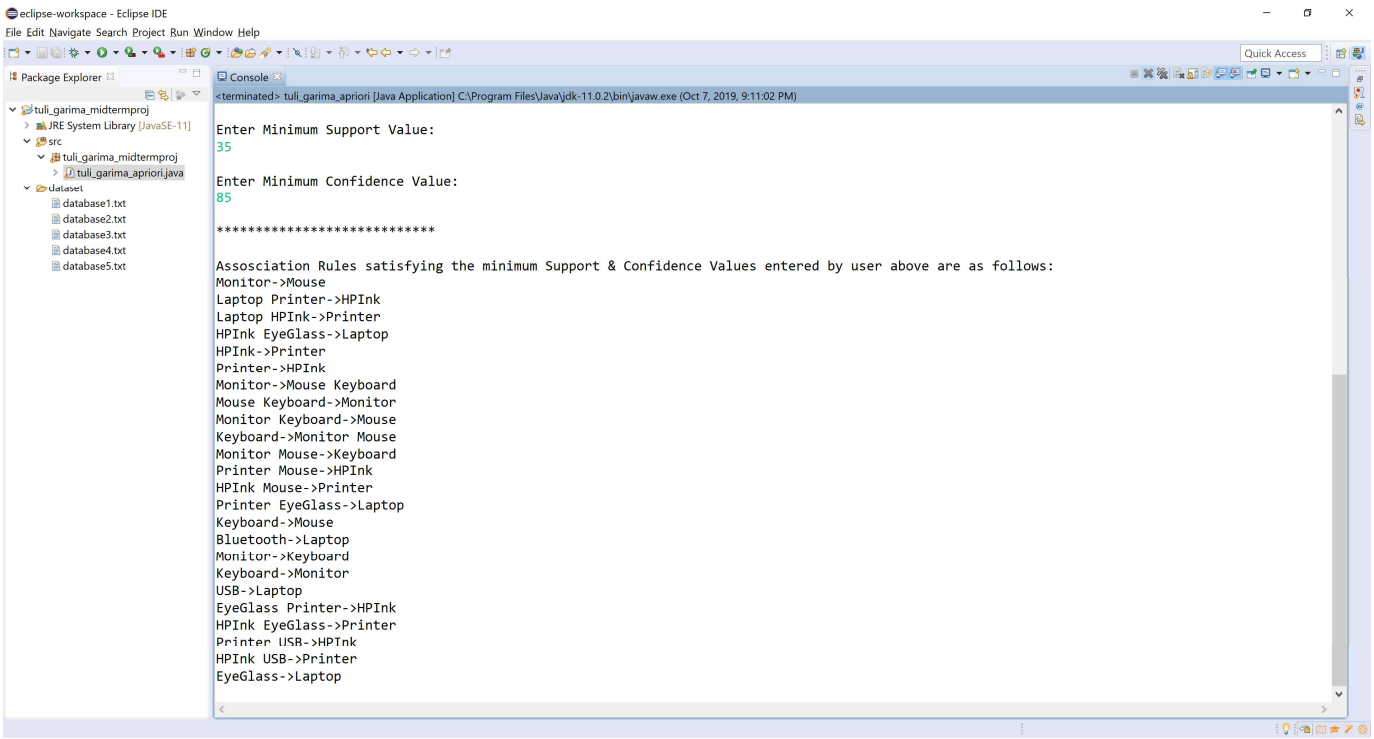



eclipse-workspace - tuli_garima_midtermproj/src/tuli_garima_midtermproj/tuli_garima_apriori.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
tuli_garima_midtermproj
JRE System Library [JavaSE-11]
src
tuli_garima_midtermproj
tuli_garima_apriori.java
dataset
database1.txt
database2.txt
database3.txt
database4.txt
database5.txt
tuli_garima_apriori.java
99
100 // Replace different file name here for different data set
101 String fileName = "dataset\\database3.txt";
102 // Local variable for each line
Console
<terminated> tuli_garima_apriori [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (Oct 7, 2019, 8:42:57 PM)
KitKat Hershey->Godiva
Cadbury->Hershey
Cadbury->Godiva
Godiva->Cadbury
Bournville->KitKat
Cadbury Godiva->Twix
Cadbury->Twix Godiva
Twix Godiva->Cadbury
Twix Cadbury->Godiva
Twix->Hershey
KitKat->Hershey
Hershey->KitKat
Cadbury->KitKat
Twix->Cadbury
Cadbury->Twix
Lindt->Hershey
Godiva Bournville->Twix
Twix Bournville->Godiva
Bournville->Twix Godiva
Twix Godiva->Bournville
Twix->Godiva
KitKat->Twix
Twix->KitKat
Twix->Bournville
Bournville->Twix
Hershey->Godiva
Godiva->Hershey
KitKat->Godiva

Screenshot of Output for Dataset 4:

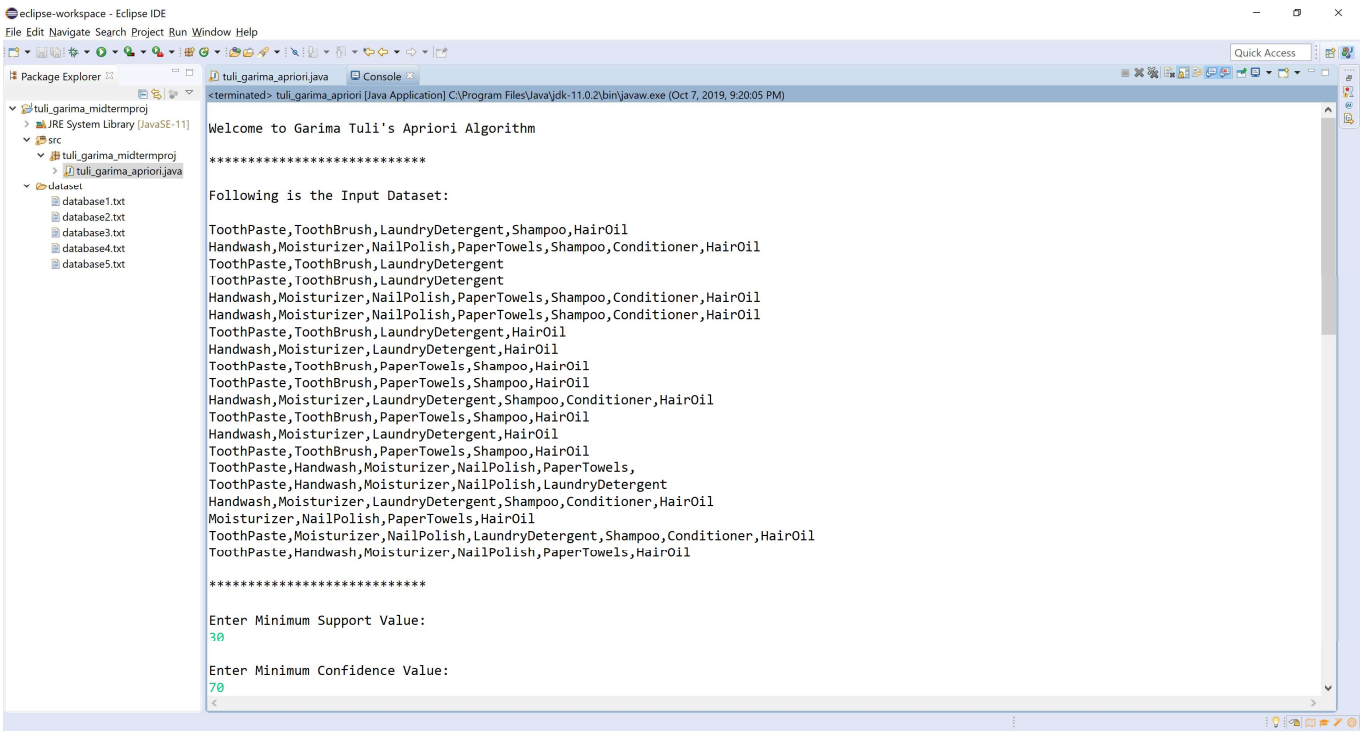
User enters minimum support value as 35 and minimum confidence value as 85 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.

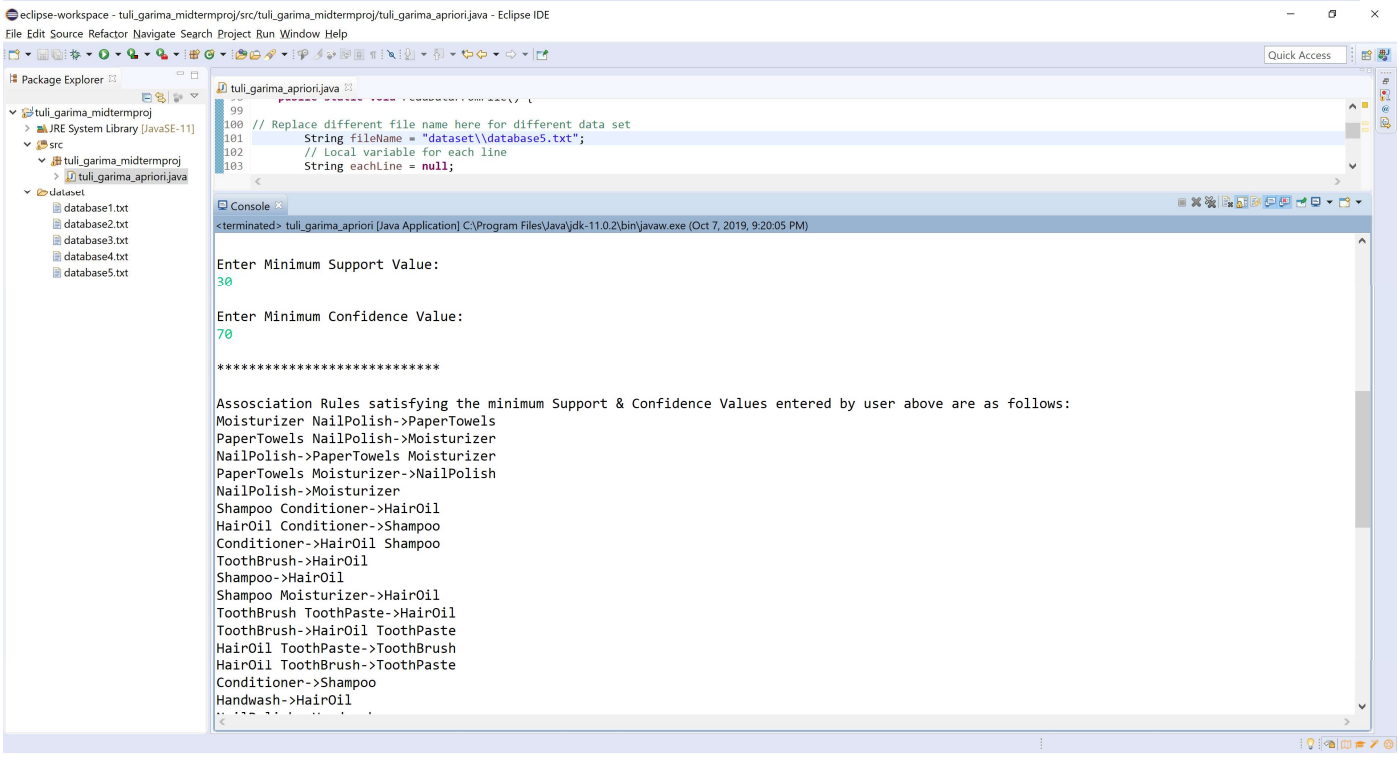


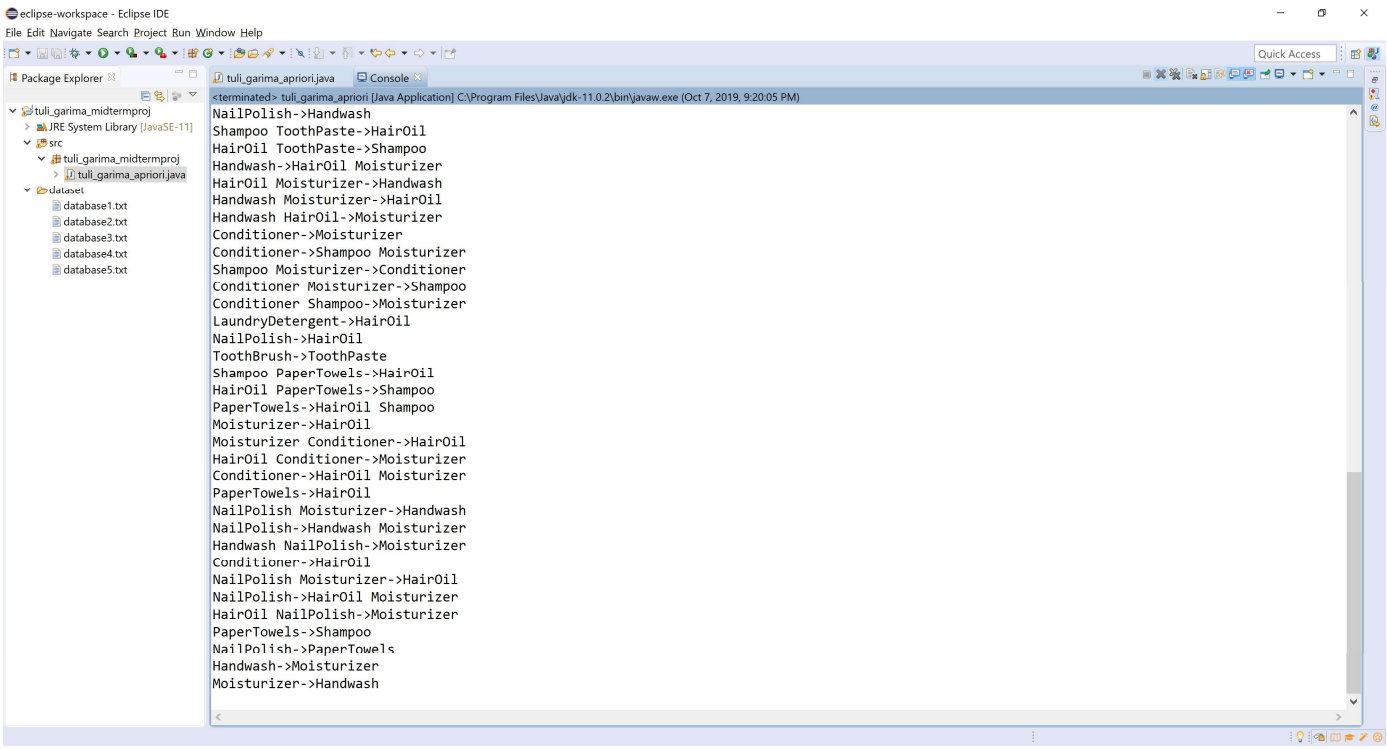


Screenshot of Output for Dataset 5:

User enters minimum support value as 30 and minimum confidence value as 70 (as displayed in green color in the screenshot below). This generates the association Rules in the output which satisfy the minimum Support & Confidence Values entered by user.







Conclusion

Thus, its observed that Apriori Algorithm generates various Association rules for 5 different datasets when user inputs different minimum support & confidence values for each dataset.