```java
package App;

public class AdminPanel {
    String id, pass;
    String c;

    AdminPanel() {
        // admin privilege starts here
        System.out.println("\tLogin Successful");

        // Use AppScanner instead of Scanner
        try {
            // 2ND PART
            System.out.println("\t═══════════════════════════════════");
            System.out.println("\t\tAdmin Panel");
            System.out.println("\t═══════════════════════════════════");
            System.out.println("\t'⟶Navigate by inserting index⟵'");
            System.out.println("\t1. Change volume of parts");
            System.out.println("\t2. Add new parts");
            System.out.println("\t3. Go back to the menu");
            System.out.println("\t═══════════════════════════════════");
            System.out.print("\tInsert a number between 1 to 3:  ");

            int x;
            while (true) {
                try {
                    x = AppScanner.nextInt();
                    break;
                } catch (Exception e) {
                    System.out.println("Invalid input. Please enter a
number.");
                }
            }

            while (x < 1 || x > 3) {
                System.out.println("\tInvalid range. \n\tPlease insert a
number between 1 to 3.");
                System.out.print("\tSelect range: ");

                while (true) {
                    try {
                        x = AppScanner.nextInt();
                        break;
                    } catch (Exception e) {
                        System.out.println("Invalid input. Please enter a
number.");
                    }
                }
            }
            System.out.println("\t———————————————————————————————");

            switch (x) {
                case 1:
```

```
                    // looks for a part. if found, shows the current volume then changes it to the target;
                    Selector.printPartAll();
                    System.out.println("\tSelect a part ID from above to change its volume.");
                    System.out.print("\tChosen index :");
                    int t;

                    while (true) {
                        try {
                            t = AppScanner.nextInt();
                            break;
                        } catch (Exception e) {
                            System.out.println("Invalid input. Please enter a number.");
                        }
                    }

                    System.out.print("\tInput the new volume : ");
                    int v;

                    while (true) {
                        try {
                            v = AppScanner.nextInt();
                            break;
                        } catch (Exception e) {
                            System.out.println("Invalid input. Please enter a number.");
                        }
                    }

                    CSVWriter.updateVolume(t, v);
                    new AdminPanel();
                    break;
                case 2:
                    // looks for the highest ID, does +1, and adds parts based on type;
                    CSVWriter.addNewPart();
                    new AdminPanel();
                    break;
                case 3:
                    // creates a new Menu instance
                    new Menu();
                    break;
                default:
                    break;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```java
package App;

public class LocationFinder {
    int n;

    public LocationFinder() {
        System.out.println("\t═══════════════════════════════════════");
        System.out.println("\t\tDealership Finder");
        System.out.println("\t═══════════════════════════════════════");
        System.out.println("\t'⟶Navigate by inserting index⟵'");
        System.out.println("\t1.IN Barishal");
        System.out.println("\t2.IN Chittagong");
        System.out.println("\t3.IN Dhaka");
        System.out.println("\t4.IN Khulna");
        System.out.println("\t5.IN Rajshahi");
        System.out.println("\t6.IN Rangpur");
        System.out.println("\t7.IN Sylhet");
        System.out.println("\t═══════════════════════════════════════");
        System.out.print("\tInsert number between 1 to 5:  ");
        // Use AppScanner for user input
        this.n = AppScanner.nextInt();
        switch (n) {
            case 1:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tBarishal Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            case 2:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tChittagong Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            case 3:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tDhaka Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

```

```
System.out.println("\t═══════════════════════════════════════");
                break;
            case 4:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tKhulna Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            case 5:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tRajshahi Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            case 6:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tRangpur Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            case 7:

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\t\tSylhet Dealership");

System.out.println("\t═══════════════════════════════════════");
                System.out.println("\tAddress : Fake address\n\tCall : Fake
number");

System.out.println("\t═══════════════════════════════════════");
                break;
            default:
                // dummy
                break;
        }
        System.out.println("\t←←←←←←←Going back to menu→→→→→→→");
        new Menu();
    }
}
```

```java
package App;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Set;
import java.util.HashSet;

public class Main{
    public static void main(String[] args) {
        new Menu();
    }

}

public class Menu {
    int x;

    public Menu() {
        System.out.println("\t═══════════════════════════════════");
        System.out.println("\t\t\tMenu");
        System.out.println("\t═══════════════════════════════════");
        System.out.println("\t'⟶Navigate by inserting index⟵'");
        System.out.println("\t1. Browse parts");
        System.out.println("\t2. Find nearest dealership");
        System.out.println("\t3. Contact");
        System.out.println("\t4. Admin login");
        System.out.println("\t═══════════════════════════════════");
        System.out.print("\tInsert number between 1 to 4:  ");

        x = AppScanner.nextInt();

        while (x < 1 || x > 4) {
            System.out.println("\tInvalid range. \n\tPlease insert a number between 1 to 4.");
            System.out.print("\tSelect range: ");
            x = AppScanner.nextInt();
        }

        System.out.println("\t———————————————————————————————");

        switch (x) {
            case 1:
                // browse parts

                System.out.println("\t═══════════════════════════════════");
                System.out.println("\t\tBrowsing Window");

                System.out.println("\t═══════════════════════════════════");
                System.out.println("\t'⟶Navigate by inserting index⟵'");
                System.out.println("\t1. Browse all ");
                System.out.println("\t2. Browse by type ");
```

```
50                 System.out.println("\t3. Browse by brand ");
51                 System.out.println("\t4. Search by name");
52                 System.out.println("\t5. Go back to menu");
53 
   System.out.println("\t═══════════════════════════════════════");
54                 System.out.print("\tInsert number between 1 to 5:  ");
55 
56                 int x;
57 
58                 while (true) {
59                     try {
60                         x = AppScanner.nextInt();
61                         AppScanner.nextLine();  // Consume the newline
   character
62                         break;
63                     } catch (Exception e) {
64                         System.out.println("Invalid input. Please enter a
   number.");
65                         AppScanner.nextLine();  // Consume the invalid
   input
66                     }
67                 }
68 
69 
   System.out.println("\t---------------------------------------");
70 
71                 switch (x) {
72                     case 4:
73                         // search using user input
74                         System.out.print("\tInput parts name(must be 1:1):
   ");
75                         String s = AppScanner.nextLine();
76 
   System.out.println("\t---------------------------------------");
77                         // Call the printPartName method from Selector
   class
78                         Selector.printPartName(s);
79                         break;
80                     case 5:
81                         // creates new Menu instance
82                         new Menu();
83                         break;
84                     default:
85                         // creates new constructor from Selector class
86                         // works when input 1 to 3
87                         new Selector(x);
88                         break;
89                 }
90                 break;
91             case 2:
92                 new LocationFinder();
93 
94                 break;
95             case 3:
96                 System.out.println("\tEmail : fake@fake.com");
```

```
                System.out.println("\tCall  : fake number");
System.out.println("\t-------------------------------------");
                new Menu();
                break;
            case 4:
                // admin panel
                new LoginVerification();
                break;

            default:
                break;
        }
    }
}



public class Selector extends Parts {
    int x;

    public Selector(int x) {
        super(0, "", "", "", 0, 0);
        this.x = x;
        switch (x) {
            case 1:
                printPartAll();
                break;
            case 2:
                printPartType();
                break;
            case 3:
                printPartBrand();
                break;
            default:
                break;
        }
        new CallBack();
    }

    public static void printPartAll() {
        String csvFilePath = "Database/PartsData.csv";

        try (BufferedReader br = new BufferedReader(new
FileReader(csvFilePath))) {
            String line;
            boolean isFirstLine = true;

            while ((line = br.readLine()) != null) {
                if (isFirstLine) {
                    isFirstLine = false;
                    continue;
                }

                Parts part = Parts.createFromCSVLine(line);
```

```
                printPart(part);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static void printPartAll(String s) {
        String csvFilePath = "Database/PartsData.csv";
        boolean found = false;

        try (BufferedReader br = new BufferedReader(new
FileReader(csvFilePath))) {
            String line;
            boolean isFirstLine = true;

            while ((line = br.readLine()) ≠ null) {
                if (isFirstLine) {
                    isFirstLine = false;
                    continue;
                }

                Parts part = Parts.createFromCSVLine(line);
                if (part.type.equalsIgnoreCase(s) ||
part.brand.equalsIgnoreCase(s) || part.name.equalsIgnoreCase(s)) {
                    printPart(part);
                    found = true;
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        if (!found) {
            System.out.println("No parts found for the specified criteria:
" + s);
        }
    }

    public void printPartType() {
        // Use AppScanner instead of direct Scanner
        System.out.println("Please select which type of product you want
to see by, inserting number(1-6)");
        System.out.println("'⟶ 1 for Engine.");
        System.out.println("'⟶ 2 for Wheels.");
        System.out.println("'⟶ 3 for Turbo.");
        System.out.println("'⟶ 4 for ECU.");
        System.out.println("'⟶ 5 for Rear Wing.");
        System.out.println("'⟶ 6 for Aero Kit.");
        System.out.print("Select range: ");
        int y = AppScanner.nextInt();

        while (y < 1 || y > 6) {
            System.out.println("Invalid range. Please insert a number
between 1 to 6: ");
```

```
            System.out.print("Select range: ");
            y = AppScanner.nextInt();
        }

        switch (y) {
            case 1:
                printPartAll("Engine");
                break;
            case 2:
                printPartAll("Wheels");
                break;
            case 3:
                printPartAll("Turbo");
                break;
            case 4:
                printPartAll("ECU");
                break;
            case 5:
                printPartAll("Rear Wing");
                break;
            case 6:
                printPartAll("Aero Kit");
                break;
            default:
                break;
        }
    }

    private void printPartBrand() {
        String csvFilePath = "Database/PartsData.csv";
        Set<String> uniqueStrings =
UniqueStringGenerator.generateUniqueStrings(csvFilePath);

        System.out.println("Unique Strings:");
        int i = 1;
        for (String str : uniqueStrings) {
            System.out.println(i++ + " for " + str);
        }

        int y = AppScanner.nextInt();
        while (y < 1 || y > (i - 1)) {
            System.out.println("Invalid range. Please insert a number
between 1 to " + (i - 1) + "");
            System.out.print("Select range: ");
            y = AppScanner.nextInt();
        }

        i = 1;
        for (String str : uniqueStrings) {
            if (i == y) {
                System.out.println("\t=====You chose " + str + "=====");
                System.out.println("=====The available parts from " + str
+ " are=====");
                System.out.println();
                printPartAll(str);
```

```
            } else {
                // do nothing
            }
            i++;
        }
    }

    public static void printPartName(String s) {
        printPartAll(s);
        new CallBack();
    }

    private static void printPart(Parts part) {
        PartPrinter.printPartInfo(part);
    }
}


public class UniqueStringGenerator {
    //checks the different brand name and act likes a dictionary that holds unique string
    public static Set<String> generateUniqueStrings(String csvFilePath) {
        Set<String> uniqueStrings = new HashSet<>();

        try (BufferedReader br = new BufferedReader(new FileReader(csvFilePath))) {
            String line;
            boolean isFirstLine = true;

            while ((line = br.readLine()) != null) {
                if (isFirstLine) {
                    isFirstLine = false;
                    continue;
                }

                String[] data = line.split(",");
                //  index 3 contains the string of interest
                String stringValue = data.length > 3 ? data[3].trim() : "";

                if (!stringValue.isEmpty()) {
                    uniqueStrings.add(stringValue);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return uniqueStrings;
    }
}

```

```java
package App;

public class Parts {
    protected int id;
    protected String name;
    protected String type;
    protected String brand;
    protected int price;
    protected int volume;

    // Constructor
    public Parts(int id, String name, String type, String brand, int price, int volume) {
        this.id = id;
        this.name = name;
        this.type = type;
        this.brand = brand;
        this.price = price;
        this.volume = volume;
    }

    // Factory method to create Parts object from CSV line
    public static Parts createFromCSVLine(String line) {
        String[] data = line.split(",");
        int id = Integer.parseInt(data[0].trim());
        String name = data[1].trim();
        String type = data[2].trim();
        String brand = data[3].trim();
        int price = Integer.parseInt(data[4].trim());
        int volume = Integer.parseInt(data[5].trim());

        if (type.equalsIgnoreCase("Engine")) {
            // Parse additional data for Engine
            int horsepower = Integer.parseInt(data[6].trim());
            return new Engine(id, name, type, brand, price, volume, horsepower);
        }
        else if(type.equalsIgnoreCase("Wheels")){
            int diameter = Integer.parseInt(data[6].trim());
            return new Wheels(id, name, type, brand, price, volume, diameter);
        }
        else if(type.equalsIgnoreCase("Turbo")){
            String boost = (data[6].trim());
            return new Turbo(id, name, type, brand, price, volume, boost);
        }
        else if(type.equalsIgnoreCase("Ecu")){
            return new Ecu(id, name, type, brand, price, volume);
        }
        else if(type.equalsIgnoreCase("Rear wing")){
            String material = (data[6].trim());
            return new RearWing(id, name, type, brand, price, volume, material);
```

```
        }
        else if(type.equalsIgnoreCase("Aero Kit")){
            String color = (data[6].trim());
            return new AeroKit(id, name, type, brand, price, volume,
color);
        }
        else {
            // For other types, create a regular Parts object
            return new Parts(id, name, type, brand, price, volume);
        }
    }
}

public class PartPrinter {

    public static void printPartInfo(Parts part) {
        System.out.println("        +------------------------------------>");
        System.out.println("        |      " +
part.getClass().getSimpleName() );
        System.out.println("        +------------------------------------>");
        System.out.println("        +------------------------------------>");
        System.out.println("        | - ID: " + part.id);
        System.out.println("        | - Name: " + part.name);
        System.out.println("        | - Type: " + part.type);
        System.out.println("        | - Brand: " + part.brand);
        System.out.println("        | - Price: " + part.price+"$");

        if (part instanceof Engine) {
            System.out.println("        | - Horse Power: " + ((Engine)
part).getHorsepower()+" BHP");
        }
        else if(part instanceof Wheels){
            System.out.println("        | - Wheel Size: " + ((Wheels)
part).getDiameter()+" inches");
        }
        else if(part instanceof Turbo){
            System.out.println("        | - Boost max: " + ((Turbo)
part).getBoost()+" psi");
        }
        else if(part instanceof RearWing){
            System.out.println("        | - Wing material: " + ((RearWing)
part).getMaterial());
        }
        else if(part instanceof AeroKit){
            System.out.println("        | - Color available: " +
((AeroKit) part).getColor());
        }
        System.out.println("        | - Available parts: " + part.volume);
        System.out.println("        +------------------------------------>");
    }
}

package App;

public class CallBack {
```

```
    public CallBack() {
        try {
            System.out.println("\tInstert 1 to find the Nearest dealer\n\t\t2 to go back to menu");
            System.out.print("\tInsert here: ");
            int n;
            while (true) {
                try {
                    n = AppScanner.nextInt();
                    break;
                } catch (Exception e) {
                    System.out.println("Invalid input. Please enter a number.");
                }
            }

            if (n == 1) {
                new LocationFinder();
            } else if (n == 2) {
                new Menu();
            } else {
                while (n ≠ 1 && n ≠ 2) {
                    System.out.println("\tInvalid range. \n\tPlease insert either 1 or 2.");
                    System.out.print("\tSelect range: ");

                    while (true) {
                        try {
                            n = AppScanner.nextInt();
                            break;
                        } catch (Exception e) {
                            System.out.println("Invalid input. Please enter a number.");
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

//java file for writing on CSV logic
package App;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
```

```
149
150 public class CSVWriter {
151     private static final String CSV_FILE_PATH = "Database/PartsData.csv";
152
153     // Method to update the volume of a part in the CSV file
154     public static void updateVolume(int partId, int newVolume) {
155         try {
156             List<String> lines =
    Files.readAllLines(Paths.get(CSV_FILE_PATH));
157
158             // Adjust the part index based on the starting ID (1000 in
    your case)
159             int partIndex = partId - 999;
160
161             if (partIndex >= 0 && partIndex < lines.size()) {
162                 String[] parts = lines.get(partIndex).split(",");
163                 parts[5] = String.valueOf(newVolume);
164                 lines.set(partIndex, String.join(",", parts));
165
166                 Files.write(Paths.get(CSV_FILE_PATH), lines);
167                 System.out.println("Volume updated successfully!");
168             } else {
169                 System.out.println("Invalid part ID.");
170             }
171         } catch (IOException e) {
172             e.printStackTrace();
173         }
174     }
175
176     public static void addNewPart() {
177         // Read existing lines to find the highest index
178         int highestIndex = findHighestIndex();
179         int newIndex = highestIndex + 1;
180
181         System.out.println("Adding a new part with index: " + newIndex);
182
183         // Gather information from the user
184         System.out.print("Enter part name: ");
185         String name = AppScanner.nextLine();
186
187         System.out.print("Enter part type: ");
188         String type = AppScanner.nextLine();
189
190         System.out.print("Enter part brand: ");
191         String brand = AppScanner.nextLine();
192
193         System.out.print("Enter part price: ");
194         int price = AppScanner.nextInt();
195
196         System.out.print("Enter part volume: ");
197         int volume = AppScanner.nextInt();
198
199         String additionalInfo = "";
200
201         // Based on part type, ask for additional information
```

```
202         switch (type.toLowerCase()) {
203             case "engine":
204                 System.out.print("Enter horsepower: ");
205                 additionalInfo = AppScanner.nextLine();
206                 break;
207             case "wheels":
208                 System.out.print("Enter diameter: ");
209                 additionalInfo = AppScanner.nextLine();
210                 break;
211             case "turbo":
212                 System.out.print("Enter boost: ");
213                 additionalInfo = AppScanner.nextLine();
214                 break;
215             case "ecu":
216                 // No additional information for ECU
217                 break;
218             case "rear wing":
219                 System.out.print("Enter material: ");
220                 additionalInfo = AppScanner.nextLine();
221                 break;
222             case "aero kit":
223                 System.out.print("Enter color: ");
224                 additionalInfo = AppScanner.nextLine();
225                 break;
226             default:
227                 System.out.println("Invalid part type.");
228                 return;
229         }
230 
231         // Construct the new part line
232         String newPartLine = newIndex + "," + name + "," + type + "," +
    brand + "," + price + "," + volume + "," + additionalInfo;
233 
234         // Append the new line to the CSV file
235         try (PrintWriter writer = new PrintWriter(new BufferedWriter(new
    FileWriter(CSV_FILE_PATH, true)))) {
236             writer.println(newPartLine);
237             System.out.println("New part added successfully.");
238 
239         } catch (IOException e) {
240             System.out.println("Error writing to the CSV file.");
241             e.printStackTrace();
242         }
243     }
244 
245     private static int findHighestIndex() {
246         int highestIndex = 0;
247 
248         try (BufferedReader br = new BufferedReader(new
    FileReader(CSV_FILE_PATH))) {
249             String line;
250             boolean isFirstLine = true;
251 
252             while ((line = br.readLine()) != null) {
253                 if (isFirstLine) {
```

```
                    isFirstLine = false;
                    continue;
                }

                String[] parts = line.split(",");
                int currentId = Integer.parseInt(parts[0]);
                highestIndex = Math.max(highestIndex, currentId);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return highestIndex;
    }
}

package App;

public class LoginVerification {

    public LoginVerification() {
        boolean loginSuccessful = false;

        do {
            System.out.print("\tPlease input User Id: ");
            String id = AppScanner.nextLine();
            System.out.print("\tPlease input Password: ");
            String pass = AppScanner.nextLine();

            // Default login admin admin
            if (id.equals("admin") && pass.equals("admin")) {
                // Login successful
                loginSuccessful = true;
                new AdminPanel();
            } else {
                System.out.println("\tPress R to retry or M to go back to
menu :");
                String c = AppScanner.nextLine();

                if (c.equalsIgnoreCase("M")) {
                    new Menu();
                    break;
                }
            }
        } while (!loginSuccessful);
    }
}


```

```java
package App;

import java.util.Scanner;

public class AppScanner {
    private static final Scanner scanner = new Scanner(System.in);

    public static String nextLine() {
        return scanner.nextLine();
    }

    public static int nextInt() {
        while (true) {
            try {
                return Integer.parseInt(scanner.nextLine());
            } catch (NumberFormatException e) {
                // Do nothing or handle the exception as needed
            }
        }
    }
}

public class AeroKit extends Parts {
    private String color;  // Additional attribute for Aero Kit

    public AeroKit(int id, String name, String type, String brand, int price, int volume, String color) {
        super(id, name, type, brand, price, volume);
        this.color = color;
    }

    public String getColor() {
        return color;
    }
}

public class Ecu extends Parts {
    // No additional attributes specific to ECU

    public Ecu(int id, String name, String type, String brand, int price, int volume) {
        super(id, name, type, brand, price, volume);
    }
}

public class Engine extends Parts {
    private int horsepower;

    public Engine(int id, String name, String type, String brand, int price, int volume, int horsepower) {
        super(id, name, type, brand, price, volume);
        this.horsepower = horsepower;
    }
```

```
51
52      public int getHorsepower() {
53          return horsepower;
54      }
55  }
56
57  public class RearWing extends Parts {
58      private String material;  // Additional attribute for Rear Wing
59
60      public RearWing(int id, String name, String type, String brand, int
    price, int volume, String material) {
61          super(id, name, type, brand, price, volume);
62          this.material = material;
63      }
64
65      public String getMaterial() {
66          return material;
67      }
68  }
69
70  public class Turbo extends Parts {
71      private String boost;  // Additional attribute for Turbo
72
73      public Turbo(int id, String name, String type, String brand, int price,
    int volume, String boost) {
74          super(id, name, type, brand, price, volume);
75          this.boost = boost;
76      }
77
78      public String getBoost() {
79          return boost;
80      }
81  }
82
83  public class Turbo extends Parts {
84      private String boost;  // Additional attribute for Turbo
85
86      public Turbo(int id, String name, String type, String brand, int price,
    int volume, String boost) {
87          super(id, name, type, brand, price, volume);
88          this.boost = boost;
89      }
90
91      public String getBoost() {
92          return boost;
93      }
94  }
95
96
```