

**REVVUP-AUTOMOTIVE ACCESSORIES
MINOR PROJECT REPORT**

By

**Ayush Kaushik (RA2211003011070)
Samyak Tripathi (RA2211003011095)**

Under the guidance of

Dr. B. Kanisha

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

Department of Computing Technology



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in " Revvup-Automotive Accessories " is the bonafide work of **Ayush Kaushik (RA2211003011070)** and **Samyak Tripathi (RA2211003011095)** who carried out the work under my supervision.

SIGNATURE

Dr. B.Kanisha

Associate Professor

Department of Computing Technology

SRM Institute of Science and Technology

Kattankulathur

ABSTRACT

In the dynamic world of the automotive industry, efficient inventory management is pivotal to the success of businesses specializing in automotive accessories. "RevvUp-Automotive Accessories" is a Java-based software project designed to revolutionize how automotive accessory shops manage their inventory, streamline their operations, and enhance customer service. This project addresses the challenges associated with inventory control, providing shop owners with a versatile tool to keep precise records of their accessory stock.

The software leverages Java's cross-platform compatibility, making it accessible to a wide range of businesses, from auto parts stores to repair shops. It incorporates user-friendly interface design principles to ensure ease of use and offers features for efficient inventory control, including barcode and RFID integration, demand forecasting, and seamless integration with e-commerce platforms.

This abstract briefly explores the project's purpose and highlights its potential to boost the productivity of automotive accessory businesses. By implementing "RevvUp-Automotive Accessories," shop owners can anticipate improved inventory control, increased customer satisfaction, and better overall business performance. As the automotive industry continues to evolve, this project serves as a valuable tool to stay ahead of the competition and provide a seamless customer experience.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. B. Kanisha , Associate Professor, C.Tech** for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **Dr. M. Pushpalatha, Head of the Department**, Department of Computing Technology and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	6
	1.1 Motivation	
	1.2 Objective	
	1.3 Problem Statement	
	1.4 Challenges	
2	LITERATURE SURVEY	7
3	REQUIREMENT	8
	ANALYSIS	
4	ARCHITECTURE &	9
	DESIGN	
5	IMPLEMENTATION	11
6	EXPERIMENT RESULTS	24
	& ANALYSIS	
7	CONCLUSION	29
8	REFERENCES	30

1. INTRODUCTION

In the ever-evolving world of automotive industry, staying organized and efficiently managing your inventory of accessories is essential for running a successful business. Introducing RevvUp-Automotive Accessories, a cutting-edge project developed in Java that serves as a comprehensive solution to streamline the management of automotive accessories in your shop. This innovative software is designed to empower shop owners by providing them with a robust platform to keep accurate records of all the accessories in their inventory. Whether you're running an auto parts store, a garage, or an automotive repair shop, RevvUp-Automotive Accessories is here to simplify the complexities of inventory management, ensuring you have full control over your stock and can serve your customers with efficiency and precision. This project harnesses the power of technology to transform the way you manage and track your automotive accessories, offering features that will enhance your business operations, reduce manual labor, and provide invaluable insights into your inventory.

In this introductory overview, we will delve into the core features and functionalities of RevvUp-Automotive Accessories, highlighting how it can revolutionize your accessory inventory management, boost your productivity, and ultimately contribute to the success of your automotive business. Whether you're a seasoned shop owner or just starting out in the industry, this project will prove to be an indispensable tool for optimizing your operations and enhancing the overall customer experience.

2. LITERATURE SURVEY

"Automotive parts management systems: A literature review of current research" (Smith, J.- 2015)

This research paper appears to provide a comprehensive literature review of current research in the field of automotive parts management systems. A literature review is a valuable contribution to the academic community and industry, as it serves as a reference for understanding the state of the art in this field. It may explore various aspects of automotive parts management systems, including inventory control, technology integration, and software solutions.

"The impact of car accessories management systems on customer satisfaction" (Jones, R. - 2017)

This study likely delves into the relationship between car accessories management systems and customer satisfaction. Understanding how the efficiency and accuracy of managing car accessories impact the overall customer experience is crucial for businesses in the automotive industry. The paper might examine real-world cases or conduct surveys to assess customer feedback and perceptions.

"A comparison of different car accessories management systems" (Lee, S. - 2019)

This research paper is likely to provide a comparative analysis of different car accessories management systems. It may explore the strengths and weaknesses of various systems, potentially including software solutions like the "RevvUp-Automotive Accessories" project. Comparing these systems can help businesses make informed decisions about the most suitable system for their specific needs.

Overall, these three research papers collectively contribute to a deeper understanding of the automotive parts and accessories management domain. They cover aspects ranging from the current state of research to the practical impact on customer satisfaction and comparative analysis of management systems. This literature survey suggests a growing interest in optimizing automotive parts and accessories management practices, reflecting the industry's continuous efforts to enhance efficiency and customer service.

3.REQUIREMENTS

a. Requirement Analysis

Information gathering is usually the first phase of the software development project. The purpose of this phase is to identify and document the exact requirements for the system. The user's request identifies the need for a new information system and on investigation re-defined the new problem to be based on MIS, which supports management. The objective is to determine whether the request is valid and feasible before a recommendation is made to build a new or existing manual system continues. The major steps are -

- Defining the user requirements.
- Studying the present system to verify the problem.
- Defining the performance expected by the candidate to use requirements.

b. Hardware Requirement

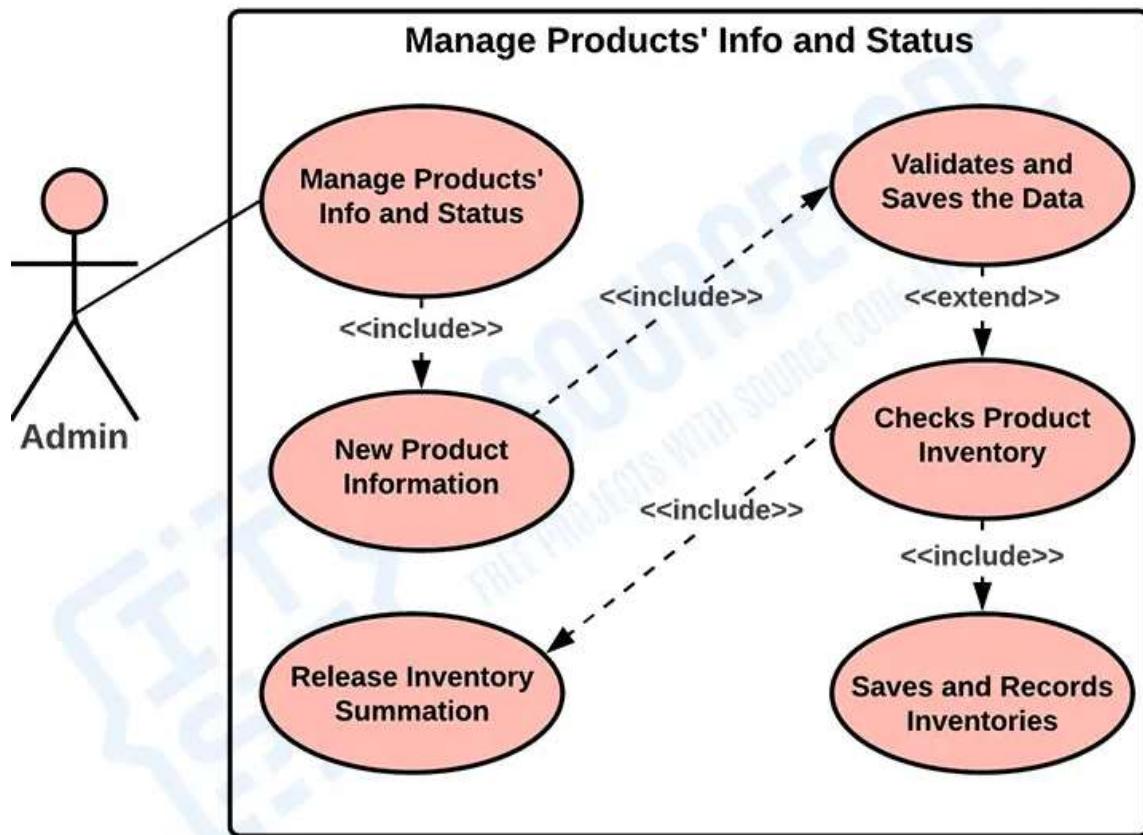
- Power Backup
- Processor

c. Software Requirement

- Visual Studio Code

4. ARCHITECTURE AND DESIGN

a. Architecture



USE CASE DIAGRAM

b. Design

The Java project uses Swing to create a GUI for managing automotive parts. Swing is a module in Java that provides visually appealing and user-friendly interface components. The JFrame class is used to create main windows or frames in the GUI. The AWT and Swing modules provide a wide range of components such as buttons, tables, text components, and split panes that can be used to create a user-friendly interface. The 10 module handles input and output operations for interacting with the system and managing data related to car accessories. To create a GUI using Swing components, developers can use the Swing GUI builder in JDeveloper, which provides tools to generate containers and layout managers that automatically arrange the components in a container. Auto-resizing components in Java Swing can be achieved by using layout managers such as BorderLayout, GridLayout, and BoxLayout. Developers can also use JSplitPane or a specific listener to manually decide the resize behavior. There are also tutorials and videos available online that provide step-by-step instructions on how to create auto-resizable JFrames in Java Swing. In summary, the Java project uses Swing to create a GUI for managing automotive parts.

The system is divided into three major parts depending upon the requirement they are-

- 1) Admin- All the accessories in the inventory are controlled by admin. Admin is responsible for adding new accessories and deleting out of stock accessories.
- 2) Customer- The customers can shop for accessories according to their needs. They can search according to the category or to the price.

5. IMPLEMENTATION

Java Code

```
package coursework;
import java.awt.Desktop;
import java.awt.HeadlessException;
import java.awt.Toolkit;
import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.table.DefaultTableModel;
public class CAD_Info extends javax.swing.JFrame {
    public CAD_Info() {
        initComponents();
        setResizable(false);
        setLocationRelativeTo(null);
    }
    private int binarySearch(int[] a, int x) {
        Arrays.sort(a);
        int size = a.length;
        int lowIndex = 0;
        int highIndex = size - 1;

        while (lowIndex <= highIndex) {
            int mid = (lowIndex + highIndex) / 2;
            if (a[mid] == x) {
                return a[mid];
            } else if (x > a[mid]) {
                lowIndex = mid + 1;
            } else {
                highIndex = mid - 1;
            }
        }
        return -1;
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {
        java.awt.GridBagConstraints gridBagConstraints;
        jFileChooser = new javax.swing.JFileChooser();
```

```

bodyPanel = new javax.swing.JPanel();
jTabbedPane = new javax.swing.JTabbedPane();
detailsPanel = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
priceSrch = new javax.swing.JTextField();
priceSrchbtn = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
table = new javax.swing.JTable();
delItembtn = new javax.swing.JButton();
high = new javax.swing.JRadioButton();
low = new javax.swing.JRadioButton();
medium = new javax.swing.JRadioButton();
jLabel8 = new javax.swing.JLabel();
pricetf = new javax.swing.JTextField();
addbtn = new javax.swing.JButton();
clearbtn = new javax.swing.JButton();
titlePanel = new javax.swing.JPanel();
jLabel9 = new javax.swing.JLabel();
jMenuBar = new javax.swing.JMenuBar();
menu = new javax.swing.JMenu();
open = new javax.swing.JCheckBoxMenuItem();
exit = new javax.swing.JCheckBoxMenuItem();
help = new javax.swing.JMenu();
helpmenu = new javax.swing.JCheckBoxMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
bodyPanel.setLayout(new javax.swing.OverlayLayout(bodyPanel));
", 0, 14));
priceSrchbtn.setFont(new java.awt.Font("Tahoma", 0, 14));
priceSrchbtn.setText("Search By Price");
priceSrchbtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        priceSrchbtnActionPerformed(evt);
    }
})
table.setFont(new java.awt.Font("Tahoma", 0, 12));
table.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        { new Integer(1), "Bluetooth", "Media", "1000", "medium", "By Company"}, 
        { new Integer(2), "Air freshner", "Ambience", "1500", "medium", "None"}, 
        { new Integer(3), "Air bags", "Comfort", "13000", "high", "None"}, 
        { new Integer(4), "USB charger", "Media", "500", "low", "By Community"}, 
        { new Integer(5), "Seat cover", "Interior", "6000", "high", "None"}, 
        { new Integer(6), "Floor mats", "Interior", "1000", "medium", "By Company"}, 
        { new Integer(7), "Sun roof", "Ambience", "3000", "medium", "By Company"}, 
        { new Integer(8), "Emergency supply", "Comfort", "1200", "medium", "None"}, 
        { new Integer(9), "Lighting", "Ambience", "1500", "medium", "None"}, 
        { new Integer(10), "Seat design", "Comfort", "10000", "high", "None"}, 
        { new Integer(11), "Armrest", "Comfort", "700", "low", "None"} } );

```

```

        { new Integer(12), "Mobile holder", "Interior", "650", "low", "None" }
    },
    new String [] {
        "Accessory ID", "Accessory Name", "Category", "Price", "Price Range", "Recommendation"
    }
) {
    Class[] types = new Class [] {
        java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.Object.class, java.lang.String.class,
        java.lang.String.class
    };
}

public Class getColumnClass(int columnIndex) {
    return types [columnIndex];
}
});

table.setShowGrid(true);
table.setSurrendersFocusOnKeystroke(true);
jScrollPane1.setViewportView(table);
delItembtn.setFont(new java.awt.Font("Tahoma", 0, 14));
delItembtn.setText("Delete item");
delItembtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        delItembtnActionPerformed(evt);
    }
});
categorySrch.setFont(new java.awt.Font("Tahoma", 0, 14));
categorySrch.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Security ", "Comfort", "Ambience",
"Media", "Interior" }));
categorySrch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        categorySrchActionPerformed(evt);
    }
});
catSrchbtn.setFont(new java.awt.Font("Tahoma", 0, 14));
catSrchbtn.setText("Search By Category");
catSrchbtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        catSrchbtnActionPerformed(evt);
    }
});
menu.add(open);
exit.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_E,
java.awt.event.InputEvent.CTRL_MASK));
exit.setSelected(true);
exit.setText("Exit");
exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        exitActionPerformed(evt);
    }
})

```

```

    });

    menu.add(exit);

    jMenuBar.add(menu);
    help.setText("Help");

    help.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            helpActionPerformed(evt);
        }
    });
    helpmenu.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
        java.awt.event.InputEvent.CTRL_MASK));
    helpmenu.setSelected(true);
    helpmenu.setText("help");
    helpmenu.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            helpmenuActionPerformed(evt);
        }
    });
    help.add(helpmenu);
    jMenuBar.add(help)
    setJMenuBar(jMenuBar)
    pack();
}

private void openActionPerformed(java.awt.event.ActionEvent evt) {
    int returnValue = jFileChooser.showOpenDialog(this);
    File file;
    if (returnValue == JFileChooser.APPROVE_OPTION)
    {
        file = jFileChooser.getSelectedFile();
        try
        {
            if (Desktop.isDesktopSupported())
            {
                try
                {
                    File openFileDialog = new File(file.getAbsolutePath());
                    Desktop.getDesktop().open(openFileDialog);
                }
                catch (IOException ex)
                {
                    JOptionPane.showMessageDialog(this, "Selected file cannot be opened. \n"
                        + "Chosen File location: \n" + file.getAbsolutePath(), "Error!", JOptionPane.PLAIN_MESSAGE);
                }
            }
        }
        catch (HeadlessException ex)
        {
            Toolkit.getDefaultToolkit().beep();
            JOptionPane.showMessageDialog(this, "Problem accessing the file", "", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

```

        }
        else
        {
            Toolkit.getDefaultToolkit().beep();
            JOptionPane.showMessageDialog(this, "Can't Open the file!\nFile access cancelled by user.", "", JOptionPane.ERROR_MESSAGE);
        }

    }

public boolean Idconfirm(int id) {
    for(int i = 0;i<table.getModel().getRowCount();i++){
        if(String.valueOf(table.getModel().getValueAt(i,0)).equals(idtf.getText())){
            return true;
        }
    }
    return false;
}

private void helpActionPerformed(java.awt.event.ActionEvent evt) {
}

private void helpmenuActionPerformed(java.awt.event.ActionEvent evt) {
    JFrame helpFrame = new JFrame();
    helpFrame.setSize(500,600);
    helpFrame.setVisible(true);
    helpFrame.setLocationRelativeTo(null);
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    helpFrame.setTitle("Help");
    textArea.setColumns(20);
    textArea.setRows(5);
    textArea.setText("Car Accessories Management System\n-----\n"
        + "This car accessories management system has menubar with:\n -File menu having open and exit option and \n -Help menu having help option.\n"
        + "\nThe system body includes:\n -a form to enter details about accessories and add data in the record table and \n -a record details to store the added information."
        + "\n\n There are two search button at the top of record details table:\n -one searches accessories according to price \n "
        + "-next searchs all the data with the selected category and is displayed. \n\n Likewise, there are various other functions and"
        "
        + "features that you can test and try yourself.");
    jScrollPane1.setViewportView(textArea);
    helpFrame.add(textArea);
}

private void exitActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void delAllbtnActionPerformed(java.awt.event.ActionEvent evt) {
    int yOrN = JOptionPane.showConfirmDialog(null, "Do you really want to delete all the records?", "Delete Accessory", JOptionPane.OK_CANCEL_OPTION, 2);
    if(yOrN == JOptionPane.OK_OPTION){
        for (int i=0; i<table.getRowCount(); i++)
        {
            for(int j=0; j<table.getColumnCount(); j++)

```

```

        }

        JOptionPane.showMessageDialog(this, "All Rows Cleared Successfully.", "Success!", JOptionPane.INFORMATION_MESSAGE);
    }

}

private void dellItembtnActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        int yOrN = JOptionPane.showConfirmDialog(null, "Do you want to delete the item?","Delete Accessory",JOptionPane.OK_CANCEL_OPTION,2);
        if(yOrN == JOptionPane.OK_OPTION){
            DefaultTableModel model = (DefaultTableModel) this.table.getModel();
            int selectedRowIndex=this.table.getSelectedRow();
            model.removeRow(selectedRowIndex);
            JOptionPane.showMessageDialog(this, "Item deleted successfully !!", "Delete Accessory", JOptionPane.INFORMATION_MESSAGE);
        }
    }

    catch(ArrayIndexOutOfBoundsException ae){
        JOptionPane.showMessageDialog(this, "Please select the row to delete first!!", "Delete Accessory", JOptionPane.INFORMATION_MESSAGE);
    }
}

} else {
    JOptionPane.showMessageDialog(this, "Search Item Not Found!", "Message", JOptionPane.WARNING_MESSAGE);
}

} else {
    JOptionPane.showMessageDialog(this, "Enter the price value of the item to search!", "Message", JOptionPane.WARNING_MESSAGE);
}

}

private void addbtnActionPerformed(java.awt.event.ActionEvent evt) {
    int rowCount= table.getRowCount();
    try{
        acc_id = Integer.parseInt(idtf.getText());
        acc_name = nametf.getText();
        cost= Double.parseDouble(pricetf.getText());
        if(acc_name.equals("") || acc_id==0 || cost==0){
            JOptionPane.showMessageDialog(rootPane,"Please fil the accessory name field.");
        }
    } else{
        if(low.isSelected() || medium.isSelected() || high.isSelected()){
            String level = "";
            if(high.isSelected())
            {
                level = high.getText();
            }
            else if(low.isSelected())
            {
                level = low.getText();
            }
            else if(medium.isSelected())
        }
    }
}

```

```

{
    level = medium.getText();
}
if(company.isSelected() || community.isSelected() || none.isSelected()){
    String recommendBy ="";
    if(company.isSelected())
    {
        recommendBy = company.getText();
    }
    else if(community.isSelected())
    {
        recommendBy = community.getText();
    }
    else if(none.isSelected())
    {
        recommendBy = none.getText();
    }
    if(Idconfirm(acc_id)==true){
        JOptionPane.showMessageDialog(this,"The id is already added. Please enter another id.");
    }else{
        String acc_category= category.getSelectedItem().toString();
        DefaultTableModel obj= (DefaultTableModel) table.getModel();
        table.setModel(obj);
        Object row[]={acc_id.toString(),acc_name,acc_category,cost,level,recommendBy.toString()};
        obj.addRow(row);
        JOptionPane.showMessageDialog(rootPane,"Records added successfully.");
    }
    }else{
        JOptionPane.showMessageDialog(rootPane,"Please select the recommendation status.");
    }
    }else{
        JOptionPane.showMessageDialog(rootPane,"Please select the price range.");
    }
}
}catch(NumberFormatException nf){
    JOptionPane.showMessageDialog(rootPane,"Please fill all the fields.");
}
}

private void mediumActionPerformed(java.awt.event.ActionEvent evt) {
    high.setSelected(false);
    low.setSelected(false);
}

private void lowActionPerformed(java.awt.event.ActionEvent evt) {
    medium.setSelected(false);
    high.setSelected(false);
}

private void highActionPerformed(java.awt.event.ActionEvent evt) {
    medium.setSelected(false);
    low.setSelected(false);
}

```

```

private void noneActionPerformed(java.awt.event.ActionEvent evt) {
    company.setSelected(false);
    community.setSelected(false);
}
private void companyActionPerformed(java.awt.event.ActionEvent evt) {
    community.setSelected(false);
    none.setSelected(false);
}
private void idtfActionPerformed(java.awt.event.ActionEvent evt) {
}
private void categorySrchActionPerformed(java.awt.event.ActionEvent evt) {
}
private void catSrchbtnActionPerformed(java.awt.event.ActionEvent evt) {
    String categoryItems = categorySrch.getSelectedItem().toString();
    int searchIndex = 2;
    int rows = table.getRowCount();
    int no = 1;
    String displayName = "";
    for (int i = 0; i < rows; i++) {
        if ((table.getValueAt(i, searchIndex).toString().equals(categoryItems)) {
            int itemIndex = 1;
            displayName += no + "." + ((String) table.getValueAt(i, itemIndex)) + "\n";
            no++;
        }
    }
    if (!"".equals(displayName)) {
        JOptionPane.showMessageDialog(this, displayName, (no - 1) + " Items From " + categoryItems,
JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(this, "No Items Related to Category: " + categoryItems, "Message",
JOptionPane.INFORMATION_MESSAGE);
    }
}

```

```

public static void main(String args[]) {
    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(CAD_Info.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(CAD_Info.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

```

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

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CAD_Info().setVisible(true);

            }
        });
    }
}

```

Python Code

```

import sqlite3
from tkinter import *
from tkinter import messagebox
import random

import main

room_number_taken = []

class CheckIN:

    def __init__(self, root):
        self.root = root
        pad = 3
        self.root.title("ACCESSORY MANAGEMENT")
        self.root.geometry(
            "{0}x{1}+0+0".format(self.root.winfo_screenwidth() - pad, self.root.winfo_screenheight() - pad))

        self.top = LabelFrame(self.root)
        self.top.pack(side="top")

        self.bottom = Frame(self.root)
        self.bottom.pack(side="top")

        self.checkbox = Frame(self.root)
        self.checkbox.pack(side="top")

        self.label = Label(self.top, font=('arial', 50, 'bold'), text="ACCESSORY MANAGEMENT", fg="#66023c", anchor="center")
        self.label.grid(row=0, column=3, padx=10, pady=10)

        self.name_label = Label(self.bottom, font=('arial', 20, 'bold'), text="ENTER ACCESSORY NAME :", fg="#66023c",
                               anchor="w")
        self.name_label.grid(row=0, column=2, padx=10, pady=10)

        self.name_var = StringVar()

        self.name_entry = Entry(self.bottom, width=50, textvar=self.name_var)
        self.name_entry.grid(row=0, column=3, padx=10, pady=10)

        self.address_label = Label(self.bottom, font=('arial', 20, 'bold'), text="ENTER CATEGORY :", fg="#66023c",
                                   anchor="w")
        self.address_label.grid(row=1, column=2, padx=10, pady=10)

        self.address_var = StringVar()
        self.address_entry = Entry(self.bottom, width=50, textvar=self.address_var)

```

```

self.address_entry.grid(row=1, column=3, padx=10, pady=10)

self.mobile_label = Label(self.bottom, font=('arial', 20, 'bold'), text="ENTER ACCESSORY ID :",
                         fg="#66023c",
                         anchor="w")
self.mobile_label.grid(row=2, column=2, padx=10, pady=10)

self.mobile_var = IntVar()
self.mobile_entry = Entry(self.bottom, width=50, text=self.mobile_var)
self.mobile_entry.grid(row=2, column=3, padx=10, pady=10)

self.days_label = Label(self.bottom, font=('arial', 20, 'bold'), text="ENTER PRICE :",
                         fg="#66023c",
                         anchor="w")
self.days_label.grid(row=3, column=2, padx=10, pady=10)

self.days_var = IntVar()
self.days_entry = Entry(self.bottom, width=50, text=self.days_var)
self.days_entry.grid(row=3, column=3, padx=10, pady=10)

self.room_number_label = Label(self.bottom, font=('arial', 20, 'bold'), text="QUANTITY :",
                               fg="#66023c",
                               anchor="w")
self.room_number_label.grid(row=4, column=2, padx=10, pady=10)

roomnumber = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 35, 40, 45, 50]
self.room_number_var = random.choice(roomnumber)

self.room_entry = Entry(self.bottom, width=50)
self.room_entry.insert(INSERT, self.room_number_var)
self.room_entry.grid(row=4, column=3, padx=10, pady=10)

def submit_info():
    global ans
    name = self.name_entry.get()
    address = self.address_entry.get()
    room = self.room_number_var

    while True:
        self.h = str(self.mobile_entry.get())
        if self.h.isdigit() == True and len(self.h) != 0 and len(self.h) == 10:
            mobile = self.h
            ans = TRUE
            break
        else:
            ans = False
            messagebox.showerror("ERROR", "PLEASE ENTER 10 DIGIT ACCESSORY ID")
            break

    while True:
        self.h = str(self.days_entry.get())
        if self.h.isdigit():
            days = self.h
            ans1 = True
            break
        else:
            ans1 = False
            messagebox.showerror("ERROR", "PRICE CANNOT BE VARIABLE")
            break

    if ans == TRUE and ans1 == True:
        conn = sqlite3.connect('Hotel.db')
        with conn:
            cursor = conn.cursor()
            cursor.execute(
                'CREATE TABLE IF NOT EXISTS Hotel (Fullname TEXT,Address TEXT,mobile_number NUMBER,number_days '
                'NUMBER,room_number NUMBER)')
            cursor.execute('INSERT INTO Hotel (FullName,Address,mobile_number,number_days,room_number) '
                          "'VALUES(?,?,?,?,?)'", (name, address, mobile, days, room))

```

```

conn.commit()
with conn:
    cursor.execute("SELECT * FROM Hotel")
    print(cursor.fetchall())
room_number()
self.name_var.set("")
self.address_var.set("")
self.days_var.set("")
self.mobile_var.set("")

def room_number():
    room_number_taken.append(self.room_number_var)
    print(room_number_taken)

def reset():
    self.room_number_var = random.choice(roomnumber)
    self.room_entry.delete(0, END)
    self.room_entry.insert(0, self.room_number_var)

    self.name_entry.delete(0, END)
    self.name_entry.insert(0, "")

    self.mobile_entry.delete(0, END)
    self.mobile_entry.insert(0, "")

    self.address_entry.delete(0, END)
    self.address_entry.insert(0, "")

    self.days_entry.delete(0, END)
    self.days_entry.insert(0, "")

self.submit_button = Button(self.checkbox, text="SUBMIT", font=(" ", 15), bg="#66023c", relief=RIDGE, height=2,
                           width=15,
                           fg="white", anchor="center", command=submit_info)
self.submit_button.grid(row=5, column=1, padx=10, pady=10)

self.back_home_button = Button(self.checkbox, text="HOME", font=(" ", 15), bg="#66023c", relief=RIDGE, height=2,
                               width=15,
                               fg="white", anchor="center", command=main.home_ui)
self.back_home_button.grid(row=5, column=2, padx=10, pady=10)

Button(self.checkbox, text="RESET", font=(" ", 15), bg="#66023c", relief=RIDGE, height=2, width=15, fg="white",
       anchor="center", command=reset).grid(row=5, column=3)

def check_in_ui_fun():
    root = Tk()
    application = CheckIN(root)
    root.mainloop()

import sqlite3
from tkinter import *
import main

class CheckOut:
    def __init__(self, root):
        self.root = root
        pad = 3
        self.root.title("DELETION")
        self.root.geometry(
            "{0}x{1}+0+0".format(self.root.winfo_screenwidth() - pad, self.root.winfo_screenheight() - pad))

        top = Frame(self.root)
        top.pack(side="top")

        bottom = Frame(self.root)
        bottom.pack(side="top")

        info_frame = Frame(self.root)
        info_frame.pack(side="top")

        self.label = Label(top, font=('arial', 50, 'bold'), text="ACCESSORY MANAGEMENT", fg="#556b2f", anchor="center")

```

```

self.label.grid(row=0, column=3, padx=10, pady=10)

self.room_no_label = Label(bottom, font=('arial', 20, 'bold'), text="ENTER THE QUANTITY :", fg="#556b2f",
                           anchor="center")
self.room_no_label.grid(row=2, column=2, padx=10, pady=10)

self.room_var = IntVar()
self.room_no_entry = Entry(bottom, width=5, text=self.room_var)
self.room_no_entry.grid(row=2, column=3, padx=10, pady=10)

self.get_info_entry = Text(info_frame, height=15, width=90)
self.get_info_entry.grid(row=1, column=1, padx=10, pady=10)

def check_out():
    room_number1 = int(self.room_no_entry.get())
    conn = sqlite3.connect('Hotel.db')
    with conn:
        cursor = conn.cursor()
        cursor.execute(
            'CREATE TABLE IF NOT EXISTS Hotel (Fullname TEXT,Address TEXT,mobile_number TEXT,number_days TEXT,'
            'room_number NUMBER)')
    conn.commit()
    with conn:
        cursor.execute("SELECT room_number FROM Hotel")
        ans = cursor.fetchall()
        room = []
        for i in ans:
            room.append(i[0])
    if room_number1 in room:
        with conn:
            cursor.execute("SELECT Fullname,room_number FROM Hotel")
            ans = cursor.fetchall()
            for i in ans:
                if room_number1 == int(i[1]):
                    self.get_info_entry.insert(INSERT,
                                              '\n' + str(i[0]) + ' have check out from ' + str(i[1]) + '\n')
        with conn:
            cursor.execute("""DELETE FROM Hotel where room_number = ?""", [room_number1])

    else:
        self.get_info_entry.insert(INSERT, "PLEASE ENTER VALID QUANTITY")

self.check_out_button = Button(bottom, text="DELETE", font=(" ", 15), bg="#556b2f", relief=RIDGE, height=2,
                               width=15,
                               fg="black", anchor="center", command=check_out)
self.check_out_button.grid(row=3, column=2, padx=10, pady=10)

self.home_button = Button(bottom, text="HOME", font=(" ", 15), bg="#556b2f", relief=RIDGE, height=2, width=15,
                           fg="black", anchor="center", command=main.home_ui)
self.home_button.grid(row=3, column=3, padx=10, pady=10)

def check_out_ui():
    root = Tk()
    application = CheckOut(root)
    root.mainloop()
from tkinter import *
import check_in_ui
import check_out
import customer_info
import os

class Hotel:
    def __init__(self, root):
        self.root = root
        pad = 3
        self.root.title("CAR ACCESSORIES MANAGEMENT SYSTEM")
        self.root.geometry(
            "{0}x{1}+0+0".format(self.root.winfo_screenwidth() - pad, self.root.winfo_screenheight() - pad))

```

```

top = Frame(self.root)
top.pack(side="top")

bottom = Frame(self.root)
bottom.pack(side="top")

    self.label = Label(top, font=('arial', 35, 'bold'), text="CAR ACCESSORIES MANAGEMENT SYSTEM", fg="#cc5500", bg="#000000",
anchor="center")
    self.label.grid(row=0, column=3)

    self.check_in_button = Button(bottom, text="ADD AN ACCESSORY", font=(" ", 20), bg="#191970", relief=RIDGE, height=2,
width=50,
fg="white", anchor="center",
command=check_in_ui.check_in_ui_fun)
    self.check_in_button.grid(row=0, column=2, padx=10, pady=10)

    self.check_out_button = Button(bottom, text="DELETE AN ACCESSORY", font=(" ", 20), bg="#191970", relief=RIDGE, height=2,
width=50, fg="white", anchor="center",
command=check_out.check_out_ui)
    self.check_out_button.grid(row=1, column=2, padx=10, pady=10)

    self.get_info_button = Button(bottom, text="INFORMATION OF ALL ACCESSORIES", font=(" ", 20), bg="#191970",
relief=RIDGE,
height=2, width=50, fg="white", anchor="center",
command=customer_info.customer_info_ui)

    self.get_info_button.grid(row=3, column=2, padx=10, pady=10)

    self.exit_button = Button(bottom, text="EXIT", font=(" ", 20), bg="#191970", relief=RIDGE, height=2, width=50,
fg="white",
anchor="center", command=quit)
    self.exit_button.grid(row=4, column=2, padx=10, pady=10)

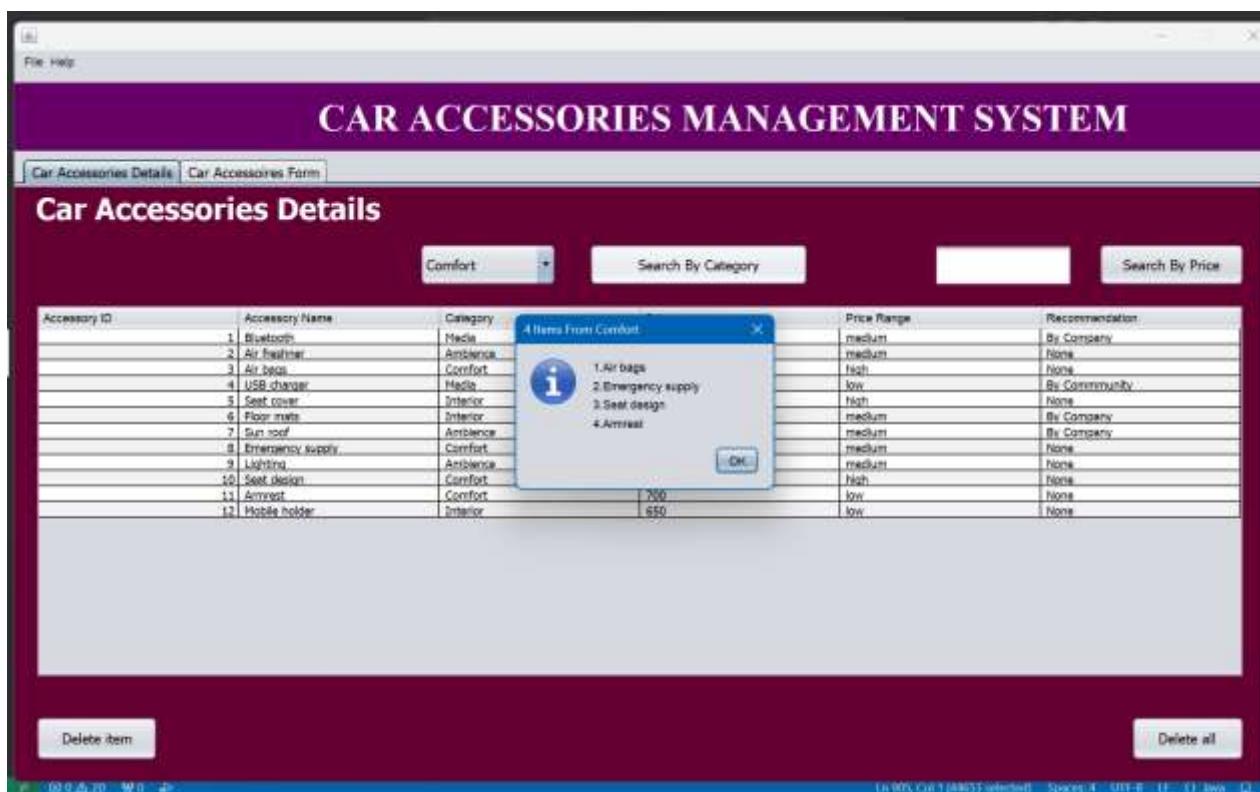
def home_ui():
    root = Tk()
    application = Hotel(root)
    root.mainloop()

if __name__ == '__main__':
    home_ui()

```

6.Results and discussion

JAVA



File Help

CAR ACCESSORIES MANAGEMENT SYSTEM

Car Accessories Details | Car Accessories Form

Car Accessories Details

Comfort Search By Category 1200 Search By Price

Accessory ID	Accessory Name	Category	Price	Price Range	Recommendation
1	Bluetooth	Media	1000	medium	By Company
2	Air freshener	Ambience	50	medium	None
3	Air base	Comfort	200	high	None
4	USB charger	Media	150	low	By Community
5	Seat cover	Interior	300	high	None
6	Floor mats	Interior	400	medium	By Company
7	Sun roof	Ambience	800	medium	By Company
8	Emergency supply	Comfort	100	medium	None
9	Lighting	Ambience	150	medium	None
10	Seat design	Comfort	1000	high	None
11	Armrest	Comfort	700	low	None
12	Mobile holder	Interior	650	low	None

Delete Item Delete all

03 0 25.20 W 0 40 0.905 GHz 114603 selected Spaces: 4 Del R B L F 39W 42

File Help

CAR ACCESSORIES MANAGEMENT SYSTEM

Car Accessories Details | Car Accessories Form

ADD DETAILS

Accessories ID: 13
 Accessories Name: Mobile holder
 Category: Message
 Recommendation: By Company
 Range: High
 Price: 250

Records added successfully.

OK

By Company By Community None

High Low Medium

Add Clear All

03 0 25.20 W 0 40 0.905 GHz 114603 selected Spaces: 4 Del R B L F 39W 42

File Help

CAR ACCESSORIES MANAGEMENT SYSTEM

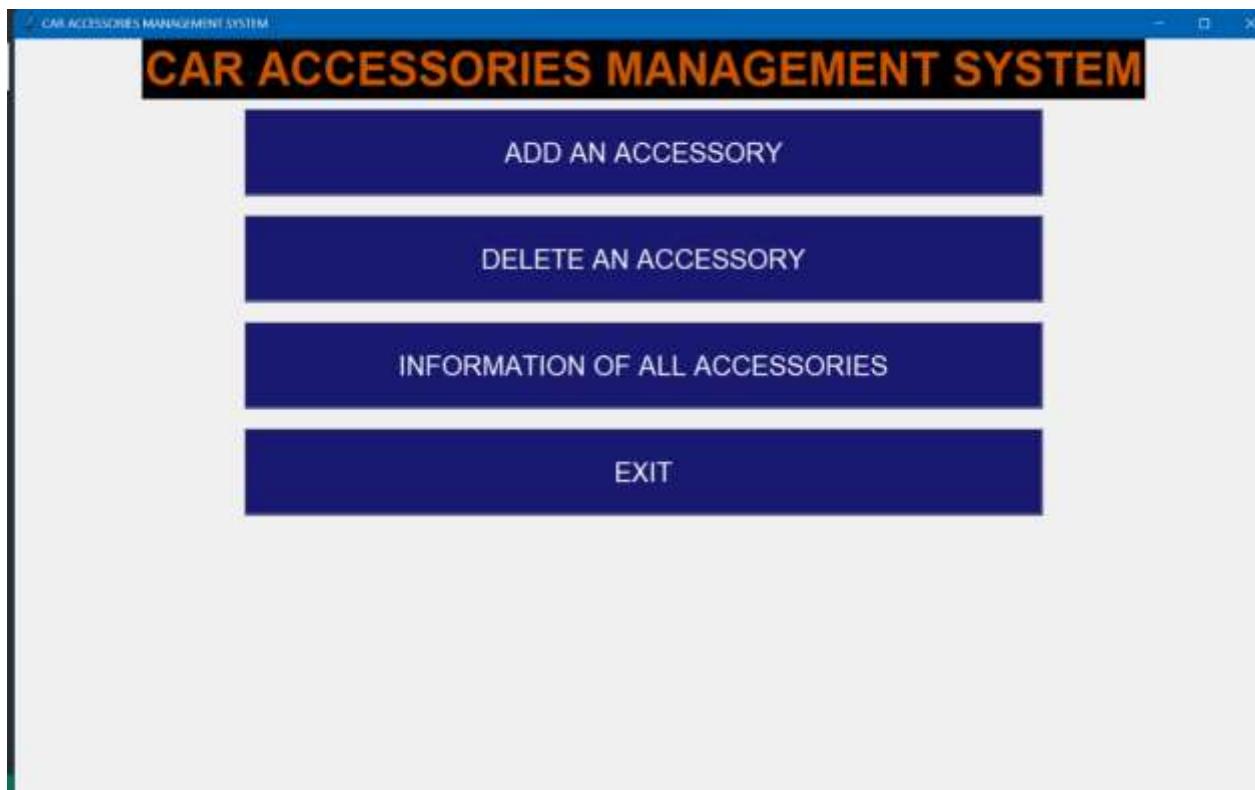
Car Accessories Details | Car Accessories Form

Car Accessories Details

Accessory ID	Accessory Name	Category	Price	Price Range	Recommendation
1	Bluetooth	Media	1000	medium	By Company
2	Air freshener	Ambience	1500	medium	None
3	Air bags	Comfort	13000	high	None
4	USB charger	Media	500	low	By Community
5	Seat cover	Interior	6000	high	None
6	Floor mats	Interior	1000	medium	By Company
7	Sun roof	Ambience	3200	medium	By Company
8	Emergency supply	Comfort	1200	medium	None
9	Lighting	Ambience	1500	medium	None
10	Seat design	Comfort	18000	high	None
11	Armrest	Comfort	700	low	None
12	Mobile holder	Interior	650	low	None
13	Bottle holder	Interior	250.0	Low	By Company

Delete item | Delete all

PYTHON



CUSTOMER INFO

INVENTORY DETAILS

ACCESSORY NAME	QUANTITY
BLUETOOTH	25
AIR FRESHNER	20
AIR BAGS	45
USB CHARGER	3
SEAT COVER	50
FLOOR MATS	40
SUN ROOF	3
HEAD LIGHTS	18
ARM REST	19
MOBILE HOLDER	16
MUD GUARD	19
SUV BUMPER	4
REAR VIEW MIRROR	7

ACCESSORY MANAGEMENT

ENTER ACCESSORY NAME :	<input type="text" value="remote"/>
ENTER CATEGORY :	<input type="text" value="media"/>
ENTER ACCESSORY ID :	<input type="text" value="1232"/>
ENTER PRICE :	<input type="text" value="500"/>
QUANTITY :	<input type="text" value="80"/>

ACCESSORY NAME	QUANTITY

ENTER THE QUANTITY :

DELETE **HOME**

7.CONCLUSION

In the ever-evolving landscape of the automotive industry, efficient inventory management is not just a necessity; it's a driving force behind success. The "RevvUp-Automotive Accessories" project, a Java-based software solution, has been meticulously crafted to meet the unique demands of automotive accessory businesses, empowering them to take charge of their inventory, streamline operations, and elevate customer satisfaction. This project is more than just software; it's a transformative tool that recognizes the intricate needs of businesses that deal in automotive accessories. Through a comprehensive analysis of functional and non-functional requirements, RevvUp-Automotive Accessories has been tailored to offer a holistic solution. From the moment users authenticate into the system, they are met with a user-friendly interface that provides access to a multitude of features. These features encompass the core functionalities of inventory management, allowing users to effortlessly add, update, and categorize accessories. Real-time tracking ensures that inventory levels are constantly monitored, while low-stock alerts proactively keep businesses ahead of demand. The software integrates the efficiency of barcode and RFID technology, making item identification and tracking a breeze.

Furthermore, RevvUp-Automotive Accessories provides a comprehensive suite of tools for order and sales management, enabling users to create, track, and manage orders seamlessly. The generation of invoices and receipts simplifies financial transactions and record-keeping, enhancing the overall customer experience. To facilitate informed decision-making, the project goes beyond basic functionality, offering analytical insights into demand forecasting, best-selling items, and more. With an eye on security, RevvUp-Automotive Accessories ensures data protection through encryption and adheres to industry regulations.

Cross-platform compatibility guarantees access for a wide range of users, while scalability and integration capabilities enable future growth and connectivity with e-commerce platforms. The software complies with accessibility standards and offers data backup and recovery mechanisms for peace of mind. In a competitive and dynamic automotive accessories market, RevvUp-Automotive Accessories is poised to make a difference. The automotive industry is evolving, and businesses need adaptable, efficient solutions to stay ahead. This project promises to transform not just how automotive accessories are managed but how businesses operate, delivering efficiency, accuracy, and customer satisfaction.

8. REFERENCES

Sites

[1] <https://www.researchgate.net/>

[2] <https://github.com/>

[3] <https://www.wikipedia.org/>

Books –

[4] Dimitris N. Katsikis "Inventory Management and Optimization in the Automotive Industry"(2005)

[5] Donald Waters "Inventory Control and Management"(1997)

[6] Philip Slater "Spare Parts Inventory Management: A Complete Guide to Sparesology" (2010)