International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol.11, No.1, 2024, pp.81-90.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

Dealer Management System

Dr.N. Baggyalakshmi [a*], G. Brindha [b], Dr.R. Revathi [c]

[a*] *Assistant Professor, Department of Computer Science, PSGR Krishnammal College for Women, Coimbatore.*
E-mail: baggyanethra@gmail.com
[b] *Student, Bsc Computer Science, PSGR Krishnammal College for Women, Coimbatore.*
E-mail: brindhaganeshan2006@gmail.com
[c] *Assistant Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore.*
E-mail: revathilakshay@gmail.com

Abstract

To keep track of all the things happening in the showroom, such as customer maintenance, generating quotations, selling vehicles, tracking customer dues for instalment customers, selling vehicle service and spare parts, keeping track of customer follow-ups, entering customer feedback forms, and employee details, a "Dealer Management System" needs to be built. The Developers behind the Dealer Management System used Visual Studio 2008 (with ASP.NET 3.5), Visual Basic for Windows, and Microsoft SQL Server Express 2005. Easy showroom maintenance is possible because to the system's efficient report generation. This Dealer Management System software can store a lot of records in the database. It's a user-friendly software which can be used for master, service, sales and report. It has an unlimited storage and application can be used in both online and offline. Reports for billing can be generated by data wise or pricewise.

Keywords: Dealer Management, Customer Quotation, Vehicle Service, Spare Part Sales.

DOI: 10.71086/IAJSE/V11I1/IAJSE1111
Received: 03 January 2024/Accepted: 29 January 2024/Published: 08 February 2024

Introduction

It is essential for every car dealership to organise and maintain this data. The dealership management system (DMS) is essential for dealers to effectively promote to and support their consumers. Managing inventory, sales, customers, credit reports, and paperwork is all done using a dealer management system (DMS) [1]. A dealer's whole company can be managed in real-time from a central platform with the help of dealer management software.

Merchants and resellers of all sizes will find this project to be a perfect fit. You can access the Bike showroom control panel from any location at any time. In the Dealer Management System, every step of selling a bike from the dealer's showroom to the consumer is detailed. The dealer's showroom will own the bike before it is sold. The posting of the bike from the fixed asset to the current asset is the essential point of this scenario [2]. The dealer has the option to manually modify the Bike business transaction type throughout the execution of the process. This allows them to specify whether the bike is new or used when selling it to the consumer.

This internship is developed using ASP.NET as front end and MYSQL SERVER as back end. This internship gives all the detailed reports about the sales process and about the service department. The Vedants Bikes was started as an authorized service centre for the company HONDA and at the beginning Vedants bikes only dealt with the spares and service. In the year of 2020 due to the dedicated and loyal work of Vedants bikes, it became an authorized dealer of Honda Company. Since then, The Vedants bikes take care of service, spares and sales of the Honda company.

From 1959 to the end of 2019, Honda produced 400 million motorcycles, making it the largest motorcycle manufacturer in the world [3]. In 2001, Honda surpassed Toyota and Nissan to become the number two car manufacturer in Japan. As of 2015, Honda ranked as the world's eighth-largest automaker [4]. Acura was the first premium brand from Honda, the pioneering Japanese automaker. When it came out in 1986, Honda was the pioneering Japanese automaker. Garden tools, marine engines, PWCs, power generators, and a host of other items are produced by Honda in addition to their flagship vehicle and motorbike enterprises.



Fig. 1: Logo of HONDA

Fig 1 shows that the In 2000, Honda unveiled their ASIMO robot, a product of their involvement in robotics and artificial intelligence research [5]. Along with the launch of the Honda H4-420 jet in 2012 and the formation of GE HONDA AERO ENGINES in 2004, they have also forayed into the aerospace industry.

Literature Review

Mushi et.al [6] Since the world's population is expanding, they will therefore be crucial in ending food insecurity and poverty. Smallholder farmers can access a variety of digital services, but the ones that are currently available are barely adequate for their needs and do not look sustainable in the context of agriculture. Enabling access to digital services throughout the entire farming cycle under one roof is necessary to realize the potential for sustainable agriculture through data management and sharing among various stakeholders. In order to address common issues facing Tanzanian smallholder farmers and those in other agriculturally deficient nations, this paper puts forth a plan for the creation of an all-inclusive digital.

Xu et.al [7] The purpose of this paper is to formulate the long-term contract between a manufacturer and a dealership by means of Stackelberg differential game models. A two-way cost-sharing contract (TWC) and a one-way cost-sharing contract (OWC) are proposed in order to enhance the low-carbon reputation of low-carbon

products. In order to entice the dealership, the manufacturer under the OWC contract offers a specific sum of subsidies. Costs under the TWC contract are split between the dealership and the manufacturer. Furthermore, as a continuation of the suggested model, dual-channel models are also examined. Upon solving and analyzing the suggested game models, several conclusions were reached. Dealership's cost coefficient in relation to impacts both the manufacturer's efforts to reduce emissions and the dealership's low-carbon promotion initiatives.

Onesmo [8] In order to comprehend the socioeconomic traits, difficulties, and roles of waste pickers in Tanzanian cities, this study examined this group. Data from waste pickers in the city was gathered using a mixed-methods research design that included questionnaires, interviews, and observations. The results showed that Arusha produces 550 tons of waste a day, with 60% of the industry's workers being men who pick waste. 90% of waste pickers were between the ages of 18 and 49 when they were actively employed, and 2/3 had only completed elementary school (66%). 42.5% of waste pickers were married, and 50% of them were single. Less than one-third had migrated, and a sizable portion (69%) were local residents. Waste pickers in Arusha make, on average, up to 153,000 TZS (US$66) per month; as a result, they are looking for other employment.

Ali et.al [9] As a result, the study's objectives are to determine and analyze any potential WD components that may be present in SWM as well as to gauge the rate at which each component is currently recovering and the overall rate of WD. Both at the landfill site and the source, MSW generation and characterization were done in order to find and measure valuable items in the waste stream. The methodology mentions using an equation to calculate the rate of waste diversion, whereas the quantification of each individual component is calculated based on characterization. According to the findings, a city produces 967.33 tons of municipal solid waste per day at a generation rate of 0.474 kg per capita. Valuable items are recovered from the waste stream in just 125.00 tons/day (13.17%), while 840 tons/day (86.81%) end up untreated at the landfill.

Kadim & Sunardi [10] A quantitative descriptive approach is used in this research. The study's population consisted of citizens from Jakarta, Bogor, Depok, Tangerang, and Bekasi (Jabodetabek), and 125 respondents were selected using a targeted sampling technique. Lisrel's data analysis. According to the study's findings, behavioral interest was positively and significantly influenced by effort expectations rather than performance expectations or social influence. Facilities conditions and behavioral interest had a positive and significant impact on Jabodetabek citizens' usage behavior of QRIS payment tools.

Prosed Methodology

Current System

By recording all actions taken in the display area, the suggested solution eliminates the drawbacks of the current setup. The application's efficiency will be enhanced by the generated reports. It will be easier to keep the service details because the service registration alerts and completion are recorded.

Advantages

- **Easiness in modification of data:** The proposed system stores customer and employee information in a fashion that allows for easy database modifications, and it efficiently manages massive amounts of data for optimal results.
- **User friendly:** Due to effective data maintenance and quick data retrieval, the suggested system is easy to use. In addition, the suggested system has a graphical user interface, so it's easy for users to navigate and operate.
- **Reports are easily generated:** In the suggested system, reports can be readily generated. Therefore, the proposed system may generate any kind of report, which aids managers in making decisions.
- **Sharing the data is possible:** A proposed system allows for the sharing of data. So long as they have permission to access the data, several users can share the same information in the current system. In addition, there is no need to transfer data between departments while interacting with each other.
- **No or very few paperwork:** There is little to no paperwork involved with the suggested approach. Computers can instantly gather all the data and provide a variety of bills and reports. The indestructible nature **of** the organization's data is ensured by its storage in a database. Since paper records are no longer necessary, the job also becomes much easier.

Since only the accounts are automated by the accounting software, the current system is not entirely automated. Because service details are not preserved through the system, it is very difficult to keep them, and searching through the lengthy lists of customer follow-ups in Excel takes more time than necessary. The existing data is kept in ledgers

and Excel sheets, which does not allow for the creation of suitable reports to aid in the management of the showroom.

Disadvantages Inability of Modification of Data

Improving the current system's inability to handle large amounts of data, storing customer information in a fashion that allows for database modifications, etc.

- **Not User friendly:** Due to inefficient data maintenance and sluggish data retrieval, the current system is not user friendly.
- **Difficulty in Reports Generation:** It takes a long time for reports to be generated in the existing system, or they are generated very slowly.
- **Manual Operator Control:** Manual operator control is there and leads to a lot of chaos and errors.
- **Lot of paperwork:** Even a modest transaction necessitates numerous forms to be filled out under the current system. In addition, any unforeseen event, like a fire in the office, has the potential to wipe out all of the company's data. The interconnected nature of the papers makes their loss much more problematic.
- **Inability of sharing the data:** Present systems do not permit data sharing. Because of this, the current system does not allow for the sharing of data across different users. In addition, the transfer of data is essential for the interaction between two divisions in a company.

Data Flow Diagram

DFD shows how the data communicates with the system. Data flow diagrams (DFDs) are great for modelling a lot of different components of a company's operations since they break down complex tasks into their component pieces in a way that analysts can understand. A data flow diagram models a system by utilising external entities that provide input data, which in turn transmit that data to a process that generates output data, which in turn passes to other processes on files. Processes can also receive data in the form of inputs.

System Specification

Software Environment

It is common practice to create a system requirements specification in reaction to a statement of requirements, such as a user requirements specification, and to utilise this document as the foundation for system design. In terms of both breadth and depth, the requirements expression differs from the system requirements specification. The latter may encompass the future system and its operating environment, but it may fail to hone many overarching ideas. Documents written in natural language have long served as the format for system needs and specifications.

Hardware Specification

The required system configuration for software development and software implementation is as follows. Hardware specification of the system that is used in this project are,

Table 1: Hardware Specification

Processor	Pentium 1V
Clock Speed	2 GHZ
Hard Disk Capacity	40GB
ROM	52 MB ROM
RAM	32 MB RAM

Software Specification

There are essentially three pieces of software needed to build an application: the platform, a front-end tool for user interaction, and a back-end tool for data storage.

Table 2: Software Specification

Platform	Windows XP
Front end tool	MS Visual Studio 2008 (ASP.NET 3.5)
Language	VB.NET
Backend tool	MS SQL Server Express 2005

Overview of Front End

The Active Server Pages (ASP.NET) framework is a set of tools for creating dynamic web applications that run on servers. Microsoft created it so that developers could create interactive websites, apps, and services for the web. This technology, which is the heir apparent to Microsoft's Active Server Pages (ASP), was initially introduced in January 2002 alongside version 1.0 of the.NET Framework. Because it is based on the Common Language Runtime (CLR), developers can use any.NET language that is compatible with ASP.NET to create their code. To enable ASP.NET components to process SOAP messages, the SOAP extension framework was developed for ASP.NET.

ASP.NET Compared with Classic ASP

Because it allows developers to create pages with controls that are similar to Windows user interface controls, ASP.NET makes it easier for developers to move from developing Windows applications to developing Web applications. Similar to its Windows equivalents, a Web control (such a button or label) can have properties assigned to it and react to events in the same way. Web controls, in contrast to their Windows counterparts, are able to render themselves by generating chunks of HTML and JavaScript that are then included into the final page that is sent to the user's browser.

In contrast to more traditional Web scripting environments, such as ASP and PHP, ASP.NET promotes an event-driven GUI architecture for development of applications. This framework integrates pre-existing technologies like JavaScript with built-in features like "View State" to provide state persistence (across requests) to the stateless Web.

Custom Controls

Additionally, ASP.NET applications can have custom controls built by programmers. The code for these controls is compiled into a dynamic link library (DLL) file rather than an ASCX markup file, unlike user controls. These bespoke controls are highly versatile and may be utilised in a variety of Visual Studio projects and online apps.

User Controls

User controls are encapsulations of sections of pages which are registered and used as controls in ASP.NET, etc.

Rendering Technique

The "visited composites" rendering approach is used by ASP.NET. Initialization code is built from the template file during compilation. This code represents the original template and generates a control tree (the composite). A control class instance represents a server control, while an instance of the Literal control class contains literal text. A page-specific class is generated by combining the initialization code with user-written code, typically through the assembly of many partial classes. The page also serves as the starting point for the control tree. There are a lot of steps involved in processing actual page requests. The initialization process begins by creating a new instance of the page class and running the initialization code. This creates the first control tree, which is then normally modified by the page's methods in the phases that follow. The code has the ability to modify both the structure of the tree and the properties and methods of the individual nodes since every node in the tree is a control that is represented as an instance of a class. In the end, the rendering process involves sending a visitor to each node in the tree and instructing them to render themselves using the visitor's methods. The client receives the HTML output that has been generated.

Upon completion of the request, the control tree and the instance of the page class are discarded. since of this, inexperienced ASP.NET developers often become confused since they depend on class instance members, which are deleted with each page request/response cycle.

State Management

Web servers host ASP.NET applications, and the stateless HTTP protocol is used to access them. This means that each application that use stateful interaction must have its own system for managing state. For managing states, ASP.NET offers a number of useful services. "State" is conceptually equivalent to GUI state according to Microsoft. Applications that rely on "data state" to function properly may run into issues; for instance, a finite-state machine that undergoes lazy evaluation in the time between requests or takes a lengthy time to initialise could run into problems. Web scraping can become extremely challenging, if not impossible, when using authentication and state management in ASP.NET pages.

Application

A set of shared user-defined variables holds the application state. These are available from the time the last application instance leaves and are set and initialised when the Application on Start event occurs during the first

instance's loading. By encapsulating the application state in the Applications collection, variables in the application state can be retrieved. Variables in an application's state are named.

Session State

Several session variables set by the user remain in memory throughout the session and are used by the server to store the current status of the session. The Session collection provides access to these variables, which are exclusive to each instance of the session. You have the option to have the variables deleted automatically after a certain amount of time, regardless of whether the session ends or not. Either a cookie or the session ID encoded in the URL itself is used to preserve a client-side user session.

ASP.NET supports three modes of persistence for server-side session variables:

In-process Mode

It is the ASP.NET process itself that keeps track of the session variables. While this is the quickest option, it also deletes all variables once the ASP.NET process is recycled or terminated.

State Server Mode

A distinct Windows service is used by ASP.NET to manage the state variables. The ASP.NET engine uses.NET Remoting to access data, which causes state management to happen outside of the ASP.NET process, making ASP State slower than In-Process. By switching to this mode, ASP.NET applications can be scaled and load-balanced across numerous servers. Session variables can survive ASP.NET process shutdowns because to the state management service's independence from ASP.NET. The session state server remains a potential weak spot, nevertheless, due to the fact that it operates as a single instance. There are limitations on the types that can be saved in session variables and load-balancing is not possible for the session-state service.

SQL Server Mode

Session variables can be preserved even after an ASP.NET process exits by storing state variables in a database. Sharing sessions across servers is the key benefit of this mode, which enables the programme to distribute the workload on a cluster of servers more evenly. Among ASP.NET's session state management methods, this one is the slowest.

You can save and retrieve user values as they explore ASP.NET pages in a web application with ASP.NET session state. As far as protocols go, HTTP does not keep any state. What this implies is that a web server views each page request made via HTTP as a separate request. Previous requests' variable values are not remembered by the server. ASP.NET session state allows you to store variable values for the life of a session by identifying requests from the same browser during a defined time range as a session. All ASP.NET apps have ASP.NET session state enabled by default.

Overview of Back End

Microsoft SQL Server Express

If you're looking for a free relational database management solution, go no further than Microsoft SQL Server Express. It is comprised of a database that is designed for smaller-scale and embedded applications. The product's origins can be found in SQL Server 2000's Microsoft Database Engine (MSDE). Since SQL Server 2005 was released, the 'Express' branding has been in use.

If you're looking for a dependable data management tool that offers robust functionality, data safety, and performance for embedded application clients, light Web applications, and local data stores, go no farther than Microsoft SQL Server Express Edition (SQL Server Express). You can redistribute SQL Server Express with applications at no cost, and it is designed for easy deployment and rapid experimentation. You may rest assured that SQL Server Express, as a component of the Windows Server System, will work in tandem with your existing server investments. Databases cannot exceed 10 GB in size, and SQL Server only supports 1 CPU and 1 GB of RAM.

Express Edition is suitable for application embedding or lightweight application development.

Capabilities

Microsoft SQL Server Express is a free database management system that offers many of the same capabilities as the premium, full editions of SQL Server. But certain large-scale installations can't use it because of its technical limitations. The Express product differs in a few ways. One difference is that in SQL Server 2012 and 2008 R2 express, the maximum database capacity is 10 GB. In SQL Server Express 2005 and previous, it was 4 GB, but in

the former MSDE, it was 2 GB. The limit is per database, not including log files; however, in certain cases, users may be able to access additional data by utilising numerous linked databases.

No SQL Server Agent service

Hardware-utilization Limits

Single physical CPU, but multiple cores allowable 1 GB of RAM (runs on a system with any RAM amount, but uses only at most 1 GB) Unlike the predecessor product, MSDE, the Express product does not include a concurrent workload governor to "limit performance if the database engine receives more work than is typical of a small number of users."

SQL Server Express includes several GUI tools for database management. These include:

SQL Server Management Studio Express

SQL Server Configuration Manager

SQL Server Surface Area Configuration tool

SQL Server Business Intelligence Development Studio

The predecessor product MSDE generally lacked basic GUI management tools,

- Features available in SQL Server "Standard" and better editions but absent from SQL Server Express include:
- Analysis Services
- Integration Services
- Notification Services
- Variants

Feasibility Study

Undoubtedly, a new system is costly to create and deploy. It demands resources for the system, labour, time, and money. As a result, the suggested system requirements-based feasibility assessment becomes even more necessary. The suggested system's feasibility research needs to be executed during system analysis.

The study is done in three phases:

- Technical feasibility
- Economic feasibility
- Operational Feasibility

Technical Feasibility

Outlining the inputs, outputs, files, programmes, and procedures of the system is essential for evaluating its technical viability. The amount of data, trends, update frequency, activity cycles, etc., can be used to qualify this in order to introduce the technical system. Since the "bike showroom management system" does not necessitate any extra hardware or configuration settings for execution, it satisfies the technological feasibility.

Economic Feasibility

The costs associated with Windows apps are typically rather low. Search, reporting, and functionality for numerous users are just a few of the capabilities that cost relatively little. This means that the "Bike showroom management system" is financially viable.

Operational Feasibility

Windows is a very customisable platform that lets developers build and release a wide variety of apps quickly to millions of customers all over the world.

Research and Discussion

The goal of system design is to create a working model of a system that can be built to meet user needs by outlining its structure, functionality, data, and interface. It might be viewed as a way to apply systems theory to the process of developing products. Competencies in system analysis, architecture, and engineering dovetail and complement one another. The goal of system design is to create a system that meets the needs of a target audience, be it a consumer

or the market. A more technical specification for the system's design can be derived from the functional design and requirements, as outlined in the system design paper.

Making a mental plan and then sketching it out is what this design is all about. External design, architectural design, and detailed design are the three main components that make up this process. All of the architectural plans and specific plans that make up the interior design. The internal design describes the inner workings of the system and its processing details, whereas the external design describes the features that are visible to the outside world.

The fundamental concepts of the system design include

- Abstraction
- Structure
- Information hiding
- Modularity
- Concurrency
- Verification

Input Screen

Through the process of Input Design, user-oriented inputs are transformed into computer-based representations. Errors in data processing are most often caused by inaccurate input data. The input design has the ability to control the data operator's input errors. Data entry should be as simple, logical, and error-free as feasible when designing input.

The suggested system operates solely through menus. When it comes to interactive design, it's a powerful tool. As an added bonus, it keeps the user from picking the wrong option by making it easier for them to understand the variety of choices. You can interact with any of the entering screens. All of the end-user requirements have been carefully considered during the design process.

Some other features included are

- The form title clearly states the purpose of the form
- Adequate space is given for data entry
- Data validation is done for eliminating duplicate entries

Table Design

Integrating data handling is the overarching goal of databases. Unlike with most embedded files or apps, this one is completely natural. In order to efficiently and rapidly serve a large number of users, a database stores interconnected data with little redundancy. The overarching goal is to provide users with information that is simple, fast, cheap, and adaptable.

Multiple authorised users have access to shared data in a database setting. A database is based on the idea of an integrated collection of data that gives users centralised access to the program's data. It paves the way for data to be considered an independent resource.

While designing database, several objectives must be considered

- Controlled redundancy
- Data Independence
- More information at low cost
- Accuracy and Integrity
- Recovery from failure
- Privacy and security
- Performance

Table 3: Login Details

Field Name	Data type	Constraints	Description
username	varchar (10)	Not null	username of the admin
password	varchar (10)	Not null	password information
Date	varchar (10)	Not null	current Date

The table 3 describes the login details. The table consists of the following attributes: username, password and date.

Table 4: Item Details

Field name	Data type	Constraints	Description
Item id	int (10)	Primary key	Id of the item
Item description	Varchar (10)	Not Null	items in the store
Item price	float (30)	Not Null	price of the item
Item quantity	int (20)	Not Null	quantity of the item

The Table 4 describes the store details of the proposed system. In the table Item id is the primary key. The table consist of the following attributes Item description, Item quantity, Item Price.

Table 5: Bill Details

Field name	Data type	Constraints	Description
Bill id	int (10)	Primary key	bill id
Bill description	varchar (20)	Not Null	products information in the bill

The Table 5 describes about the bill details. In this table bill id is the primary key. The table includes the following attributes bill id, bill description.

Table 6: Customer_ Details

Field name	**Data type**	**Constraints**	**Description**
Customer name	varchar (35)	Not Null	Name of the customer.
Customer address	Varchar (35) Put the cust	Not Null	customer address
Customer phone number	Varchar (20)	Not Null	phone number of the customer

The Table 6 describes about the customer details. In this table consist of the following attributes customer name, customer address, customer phone number

Output Design

Outputs are the most important and direct source of information to the customer and management. Intelligent output design will improve the system's relationship with the user and help in decision making.

Outputs are used to make permanent hard copy of the results for later consultation. The output generated by the system is often regarded as the criteria for evaluating the performance of the system. The output design was based on the following factors.

- Usefulness determining the various outputs to be printed to the system user.
- Differentiating between the outputs to be displayed and those to be printed.
- The format for the presentation of the output.

For the proposed system, it is necessary that the output should be compatible with the existing manual reports. The outputs have been formatted with this consideration in mind. The outputs are obtained after all the phase, from the system can be displayed or can be produced in the hard copy. The hard copy is highly preferred since it can be used by the controller section for future reference and it can be used for maintaining the record.

Conclusion

Dealer Management System is used by vehicle showroom dealers and owners to effectively maintain the data regarding sales of vehicles and retrieve the available types in the store. This system has been designed in such a way that any future enhancement to it will not impact any existing data. Database is designed in such a way that it handles huge volume of data without database crash. Also, I learnt how the application works & its design in both backend and front end.

Reference

Mahardika, F., Fitriani, A., & Al'Amin, M. (2023). Testing Sistem pada Dealer Management System Service Menggunakan Metode Black Box Testing. *Hello World Jurnal Ilmu Komputer*, *2*(3), 110-119.

Sutabri, T. (2023). Analisis Manajemen Resiko Pelayanan Sistem Manajemen Dealer VIAR dengan menggunakan COBIT 5. *Indonesian Journal of Multidisciplinary on Social and Technology*, *1*(2), 168-175.

Ebrahimi, A., Bakhshizadeh, P., & Varasteh, R. (2023). A predictive analytics approach to improve the dealers-manufacturer relationship in the after-sales service network; case study in the automotive industry. *International Journal of Management Science and Engineering Management*, *18*(3), 225-235.

Preunkert, J. (2023). Primary dealer systems in the European Union. *European Journal of Sociology/Archives Européennes de Sociologie*, *64*(1), 97-121.

Sinaga, T.H., Hardiansyah, H., & Dewi, A.R. (2023). Web-Based Administration Applications for Motorcycle Dealers Using FAST Framework. *Journal of Computer Networks, Architecture and High-Performance Computing*, *5*(1), 182-197.

Mushi, G.E., Serugendo, G.D.M., & Burgi, P.Y. (2023). Data management system for sustainable agriculture among smallholder farmers in Tanzania: research-in-progress. *Information Technology for Development*, *29(4),* 558-581.

Xu, C., Jing, Y., Shen, B., Zhou, Y., & Zhao, Q.Q. (2023). Cost-sharing contract design between manufacturer and dealership considering the customer low-carbon preferences. *Expert Systems with Applications*, *213*, 118877.

Onesmo, C. (2023). Exploring the socioeconomic status, challenges, and roles of waste pickers in Arusha city's solid waste management system: An insight into their contributions in Tanzania's cities.

Ali, G., Saqib, Z., Ziad, M., & Ali, J. (2023). Identification and quantification of major components of waste diversion and their recovery rates in current waste management system in Peshawar, Pakistan. *Arabian Journal of Geosciences*, *16*(1), 34.

Kadim, A., & Sunardi, N. (2023). Financial management system (QRIS) based on UTAUT model approach in jabodetabek. *International Journal of Artificial Intelligence Research*, *6(1*), 1-8.