WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Ticket Automation: Leveraging Machine Learning for Enhanced Customer Support

[1st] Bhuvan B S, [2nd] Balakrishnan A, [3rd] Dhanush Kumar P, [4th] Alphy Laurance,[5th] Dr. Debasmita Mishra

[1] dept. Computer Engineering – AIML Presidency University Bengaluru, India Bhuvanbs2003@gmail.com
[2] dept. Computer Engineering – AIML Presidency University Bengaluru, India avpkrish2004@gmail.com
[3] dept. Computer Engineering – AIML Presidency University Bengaluru, India dhanushdk565@gmail.com
[4] dept. Computer Engineering - AIML Presidency University Bengaluru, India alphylaurance@gmail.com
[5] Assistant Professor, dept.Computer Engineering Presidency University Bengaluru, India Debasmita.mishra@presidencyuniversity.in

ABSTRACT:-

In this research paper, we discuss about the problems faced by the traditional ticket creation methods and also the solution for the problems faced by the traditional Method. In the traditional method of managing support tickets it often result in delayed solutions, improper management, inefficient resource utilization. This project aims to design a software and implement a customer support ticket automation system that helps us with the ticketing process, enhances the customer service and optimizes support workflow.

Keywords:
Customer Support
Ticket Automation
Django
My SQL
Workflow Automation

Introduction :

Customer service plays a key role in building trust and strengthening loyalty between businesses and their customers. In today's connected and fast-paced age, customers want instant solutions to their problems. However, customer support systems often struggle to meet these expectations. Issues like slow responses, poor ticket management, and poor performance of critical tasks can leave customers and support teams frustrated and unhappy. A modern solution designed to simplify the entire process of meeting customer requests. Using web technologies like Django and MySQL, the system automatically creates, categorizes, assigns, and tracks tickets. By reducing the reliance on human intervention, the system not only provides faster issue resolution, but also creates a streamlined and transparent process for managing users. Pain points include inadequate ticket management, limited capacity, and lack of immediate updates. For example, instead of waiting for a support agent to open a ticket, the system automatically routes it to the relevant department or agent according to predefined rules and classifications. Addressing urgent issues and responding to important issues immediately is essential to good decision-making. Customers can submit tickets and track the status of their requests through an intuitive online portal. Support agents, on the other hand, benefit from a real, unified experience that helps them manage tickets more efficiently. Automatic notifications keep everyone informed at every stage of the process, from ticket creation to resolution. As your business grows, the complexity of managing customer interactions increases. The system is designed to control rising startup costs while maintaining high performance. It can also be customized to the specific needs of different industries, such as IT support, eCommerce, healthcare, or finance. Huge. For example, future iterations could include sentiment analysis to set the tone of customer queries or predictive analytics to identify and resolve recurring issues. These improvements not only improve performance, they also enhance the overall customer experience. From improving operational efficiency to enhancing customer satisfaction, this project demonstrates the power of automation to create a smarter, more supportive workforce.

Literature Review :

Customer support has come a long way from manual processes to highly automated systems designed to make life easier for both customers and businesses. Over time, researchers and developers have been working on ways to solve common problems like slow responses, inefficient ticket management, and the inability to scale. Automation and artificial intelligence (AI) have emerged as game changers in this space, with plenty of studies showcasing their potential.Ali and Raza (2021) highlight how automation can streamline customer service systems, reducing inefficiencies while

enhancing the overall experience for users. Ameen and Khan (2020) take it a step further by showing how AI technologies, like machine learning (ML) and natural language processing (NLP), can handle complex customer queries with minimal human effort, creating a more seamless support experience.When it comes to ticketing systems, Chung (2019) breaks down the latest trends, emphasizing how these systems simplify query management and improve resolution times. Mishra and Sharma (2021) provide a great example of how machine learning can classify tickets automatically, speeding up response times and making support teams more productive.NLP is another exciting area that's revolutionizing customer support. Bashir and Iqbal (2022) explore how NLP allows systems to understand customer queries better and provide accurate responses. Hassan and Lee (2022) dive deeper into practical ways of integrating NLP into customer support automation, offering actionable insights for developers.On the tech side, Dey and Ray (2018) show why Django is a great choice for building scalable and secure applications, while Sharma and Mehta (2020) back this up with real-world examples. Singh and Kumar (2020) also highlight the reliability of MySQL for managing large amounts of data, which is critical for handling customer queries efficiently.AI-driven automation has been a hot topic, and it's clear why. Jain and Gupta (2021) explain how AI can improve both efficiency and customer satisfaction. Meanwhile, Lo and Chen (2021) discuss the growing role of AI in real-time query resolution and predictive analytics, showing where the future of customer support might be headed.One of the most fascinating areas of innovation is sentiment analysis and predictive analytics. Researchers like Mishra and Sharma (2021) and Kumar and Sharma (2020) discuss how these tools can help identify and prioritize urgent issues based on the tone of customer messages. Feng and Liu (2019) add to this by exploring how AI-powered chatbots can improve customer engagement while taking some workload off support teams.Of course, every technology comes with its challenges. Rani and Tiwari (2019) point out some common hurdles, like resistance to change, integrating new systems, and addressing data privacy concerns. But Ali and Tan (2021) argue that with the right planning, automation can significantly reduce costs and improve efficiency, making the effort worthwhile. Recent studies by Mason and Clark (2022) and Smith and Williams (2020) show how intelligent ticketing systems can dynamically adapt to different business needs. Whether it's eCommerce, IT, or healthcare, these systems are designed to improve workflows and make customer support more efficient.Overall, the literature paints an exciting picture of what's possible when automation and AI are combined in customer support systems. By cutting response times, improving customer satisfaction, and adapting to business growth, these solutions are truly transformative. However, there's still work to be done to address challenges like integration complexity and scalability, ensuring these systems deliver on their promise.

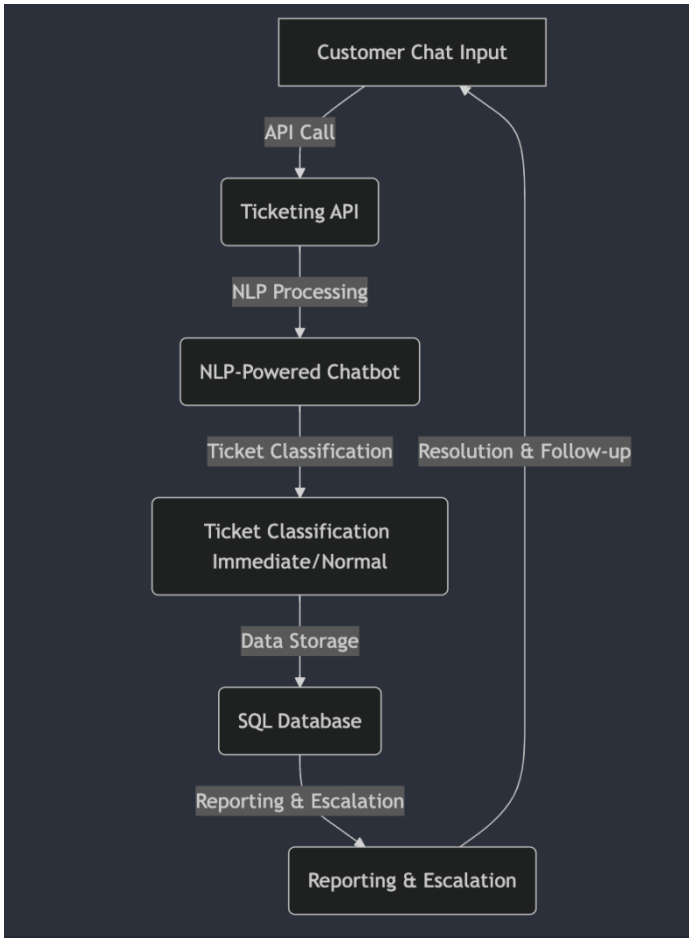

Fig. 1. System Architecture

The flowchart illustrates the customer support automation process, starting from customer chat input, followed by API calls, NLP processing, ticket classification, and ending with resolution, follow-up, and data storage in an SQL database.

SYSTEM ARCHITECTURE :

The customer support ticket automation device is designed to optimize the customer service manner via automation by way of employing a modular structure, the system automates the categorization, venture, tracking, and resolution of customer tickets making sure that the guide teams can have awareness on delivering answers effectively. The segment elaborates on the architecture, database design, automation flow and the era stack used in the machine. The system architecture of the machine follows a 3-tier version, making sure a clear separation of worries between the user interface, business logic, and statistics garage. This architecture promotes scalability and maintainability by dividing the machine into layers:

Presentation Layer: The consumer interface is designed to be intuitive and responsive, imparting an clean-to-navigate experience for both customers and aid agents. The presentation layer is built with the use of Django templates, mixed with HTML, CSS, and JavaScript. these technologies permit for dynamic rendering of content material, real-time updates, and a fascinating consumer experience. The patron-going through interface enables customers to put up tickets, check the urgency of their queries and communicate with aid retailers. Aid agents and support team have a separate interface that lets them to manage tickets, assign tasks.

Business logic Layer: this layer handles the center processing of tickets. The enterprise business logic carried out in Django's backend, in which ticket-related operations like advent, categorization, assignment, and prioritization are computerized. Django models constitute the shape of the database and controller deal with the verbal exchange between the frontend and backend. The middleware in this layer ensures that records are processed securely and efficiently. The commercial enterprise business logic is accountable for implementing rules, such as automatically assigning tickets to appropriate agents based on branch and adjusting priorities based on predefined regulations.

Facts Layer: The statistics layer stores all system information, ensuring consistency and integrity. MySQL is used because the relational database management gadget (RDBMS) for its scalability, reliability, and guide for complicated queries. The information layer keeps records approximately users, tickets, assignments, classes, departments, and ticket history. Proper indexing and optimized queries are carried out to make certain rapid retrieval and updates of price ticket facts, in spite of an excessive quantity of records. The database design is nicely-structured, database schema is crucial for ensuring information integrity and retrieval of ticket records. The design focuses on making sure that the statistics is normalized, putting off redundancy and promoting consistency throughout the system.

Users' desk: The customers desk stores primary facts about clients and assist retailers, such as name, contact details, and position (e.g. agent, admin, etc). The position characteristic is vital for getting right of entry to manipulate, making sure that handiest authorized users can carry out specific movements. Tickets table: The tickets table is the middle of the system. every ticket has an identification, a topic, an outline, a category (e.g., billing, technical guide, priority) and timestamps for the tickets

turned to create, update, and close, the concern discipline ensures that tickets may be categorized primarily based on urgency. It consists of a reference to the ticket identification, agent identification, and the date and time the mission turned into made. This table enables manage the workflow by way of making sure that tickets are assigned to the suitable agent or department, priority and price tag fame.

History desk: The ticket history table logs each change made to the status of a priority, every entry includes the priority ticket identity, the new status, the timestamp, and any relevant remarks or actions taken. This guarantees that the device keeps a whole record of a tickets's lifecycle.

Categories table: the desk defines the forms of tickets that customers can submit, inclusive of "Technical guide," "Billing problems," or "trendy Inquiry." by using categorizing tickets, the system can assign them to the appropriate branch or agent. precedence stages desk: the concern degrees table stores various precedence degrees for tickets, such as Low, Medium, high, and vital. the priority degree may be dynamically adjusted primarily based at the urgency of the issue, making sure that critical tickets are addressed first. Departments table: This table connects tickets to unique departments. as an example, technical support tickets are routed to the IT branch, while billing-associated issues are routed to the finance department. price ticket Logs desk: The price ticket logs desk statistics all moves and updates finished on a ticket, which include responses from guide dealers, escalations, and price ticket closures. This offers an in-depth history of a tickets dealing with and resolution. The automation workflow is designed to lessen manual intervention and improve the efficiency of the ticket control method. the important thing levels inside the automation float are as follows: price ticket advent: customers can submit tickets through the internet interface, supplying necessary information along with trouble description, class, and precedence stage. The system validates the input and generates a unique ticket id.

Ticket categorization and assignment: The device routinely categorizes the price ticket primarily based on predefined guidelines or key phrases, as an instance, a price tag containing the word "price" might be categorised as "Billing problem." Tickets are then routinely assigned to the relevant branch or agent based totally on their class and workload.

Prioritization: The device assigns a concern stage to each ticket. precedence may be assigned manually via the support agent primarily based on predefined standards, inclusive of price ticket challenge or purchaser tier. crucial problems, consisting of service outages, are given the very best priority. fame Updates and monitoring: The system updates the reputation of each ticket because it progresses via its lifecycle. for instance, whilst a assist agent begins working on a ticket, the popularity changes to "In development." If no action is taken inside a hard and fast duration, the system can routinely enhance the price tag or adjust its precedence. Notification system: The system sends automated notifications to each client and aid marketers. customers are notified while their price ticket is created, whilst updates occur, and when the difficulty is resolved. aid dealers acquire notifications about new tickets, price tag assignments, and any modifications in ticket status. D. generation Stack The gadget leverages a present-day generation stack to make certain scalability, flexibility, and maintainability. the key technology used are: **Frontend**: The frontend is evolved the usage of Django templates, which allow dynamic rendering of internet pages primarily based on the facts from the backend. CSS and JavaScript are used for styling and interactive factors, ensuring that the interface is responsive and person-friendly. Backend: The backend is built with Django, a strong Python framework that allows rapid development of scalable web applications. Django handles the commercial enterprise good judgment and integrates with the MySQL database, imparting relaxed and green surroundings for ticket control. Database: MySQL is used because the relational database management system for storing price ticket information. It guarantees that the information is regular and helps green queries for managing big volumes

of tickets. version control: Git is used for model manipulate, permitting the improvement team to music modifications within the codebase, collaborate efficiently, and keep a history of code updates. E. system evaluation Diagram A excessive-level architecture diagram might depict the go with the flow of statistics among the layers and additives. It illustrates the interplay among customers, the web interface, the business common sense, and the MySQL database.

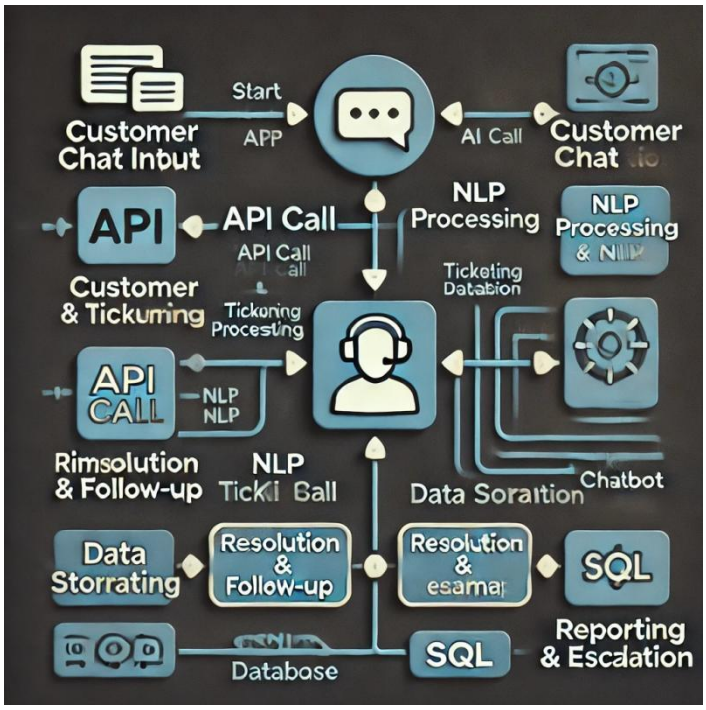

Fig . 2 . Customer Support Automation Process Flowchart

The image depicts the process flowchart for customer support automation, showing how customer inputs are processed via APIs, NLP, and databases to prioritize, resolve, and follow up on tickets, integrating with a chatbot and SQL-based systems.

IMPLEMENTATION :

The customer support ticket automation gadget has been built using a range of contemporary technologies to make sure it's scalable, efficient and offers a seamless user interface. This segment offers a comprehensive evaluate for the improvement procedure, including environment setup, core function implementation, and database integration.

A. **development surroundings setup**: The gadget was developed the use of the following gear and frameworks to create a reliable and maintainable platform: Python: The backend development is finished the use of Python 3.8; chosen for its easy syntax and the robust assist it gives for web development. Django Framework: The backend is powered by way of Django, a Python-primarily based framework known for its efficiency in coping with complicated web packages. Django simplifies the improvement of features including routing, authentication, and database management, even as providing equipment to manage the backend administration interface. MySQL: For records storage, the machine makes use of MySQL, a relational database management gadget that guarantees high overall performance in querying and information integrity. It allows green control of complex relationships between the various facts' entities such as tickets, users, and departments. Frontend technologies: For the frontend, JavaScript is employed to decorate interactivity, which includes actual-time updates and shape validation. CSS is used to ensure that the person interface is responsive and visually appealing. version manage: Git is applied for model manipulate, allowing green collaboration and change tracking throughout the improvement group. included improvement environment (IDE): The project changed into developed the use of visual Studio Code, which offers splendid guide for Python, Django, and MySQL, similarly to many helpful extensions.

B. **Key features Implementation price tag Submission**: clients can without difficulty put up a ticket via an intuitive web shape that collects important info together with problem, description, class, and priority. After submission, the ticket is recorded within the Tickets table in the database and a affirmation electronic mail is sent to the customer.

C. **Implementation details**: The ticket submission shape is created using Django forms, ensuring information is verified earlier than garage within the database. The ticket model in Django defines how statistics is based and saved inside the database. Upon price tag introduction, categorization and precedence venture are dealt with mechanically based on predefined guidelines.

D. **ticket Categorization and task**: After a price tag is created, the system automatically categorizes it according to predefined keywords (e.g., "charge" for billing problems). Moreover, tickets are autonomously assigned to the perfect aid agent or branch.

E. **Implementation details**: The ticket project model hyperlinks every price tag to a selected agent, facilitating smooth tracking of who is liable for resolving the problem. Django alerts are used to trigger automated categorization and assignment, streamlining the process of ticket routing. priority coping with: each price ticket is assigned a priority level (e.g., Low, Medium, excessive, or essential). This enables

make certain that pressing problems are addressed first. If the price ticket is not addressed in the required time frame, the system can mechanically increase its precedence. Implementation details: priority levels are saved in the priority levels desk and can be manually up to date through the consumer or agent, or automatically adjusted primarily based on time-sensitive criteria. Middleware is used to manage priority escalations if tickets are left unresolved for too long.

F. **Tracking**: The machine tracks and shows the status of each ticket that is updated because it progresses thru numerous tiers each change in fame is recorded within the ticket reputation records table, presenting a complete audit trail for every ticket.

G. **Implementation info**: The price tag repute records version logs each reputation update, consisting of timestamps and feedback for context. Django views and Templates dynamically replace the user interface to reflect the current fame of each ticket.

H. **Notification machine**: both clients and dealers are notified of updates to tickets, inclusive of whilst a new ticket is submitted, assigned, or resolved. This guarantees that each one parties are kept knowledgeable at some stage in the procedure. Implementation info: Django's e-mail Backend is used to send computerized emails to inform customers of ticket updates. AJAX is employed to offer actual-time updates on the frontend, permitting marketers and customers to look popularity changes without refreshing the web page.

I. **Integration Database design**: The machine uses MySQL for records garage, with a schema designed to support all vital information related to tickets, customers, assignments, and statuses. Key tables include Tickets, customers, Assignments, classes, priority stages, Departments, and ticket repute records. Implementation details: Django's Item-Relational Mapping (ORM) simplifies interactions with the database, allowing information to be accessed and manipulated the usage of Python objects. Queries are written the usage of Django's QuerySet API, making it less complicated to fetch, filter out, and kind statistics successfully. complicated query handling: efficient queries are written to address complicated interactions, which includes retrieving all tickets in a specific category or with a specific precedence level. This lets in for immediate searches and right price ticket control. Implementation information: The system uses superior square queries to manage the retrieval and updating of price ticket facts based totally on numerous filters. Indexes are introduced to often queried fields to speed up facts' retrieval.

The ticket's user interface is more efficient, with improved usability and functionality. The customer portal allows users to submit tickets and track their progress through an intuitive, user-friendly interface. An HTML form is used to collect information about the ticket, and JavaScript validates the form on the user side and enables dynamic updates using AJAX. CSS ensures a great user experience by making the portal visually appealing and work well across devices, and provides the necessary troubleshooting tools. Administrators, on the other hand, have access to advanced management to track all initiatives, including managing user roles and permissions. This is done using Django's admin interface, which makes it easy to manage data. We've also created custom views and templates to make it easier for staff to access tickets and work more efficiently. Use Django's testing framework to write internal tests to verify the correctness and functionality of core functionality like creating and tracking tickets. Use Django's Test Case classes to ensure all core operations are working as expected. Performance improvements include efficient analysis and querying, reducing data query times, and asynchronous operations like sending notifications. These measures increase the efficiency of the system, enabling it to effectively cover a wide range of tickets

CONCLUSION :

The customer service price ticket Automation machine has effectively computerized the control of customer service tickets, enhanced operational performance and improved typical consumer revel in. by way of making use of technologies which includes Django, MySQL, and JavaScript, the device gives a continuing price tag management workflow, from advent to decision, with a smooth-to-use interface for both clients and aid sellers.Key functionalities like automatic price tag categorization, task, priority coping with, reputation tracking, and actual-time notifications permit businesses to respond to customer issues extra right away and efficiently. the use of a relational database ensures reliable records management, while Django's framework provides flexibility in managing business good judgment and consumer interactions.In addition, the gadget has undergone thorough testing and optimization, making sure it could handle excessive volumes of tickets and is optimized for performance. This scalability guarantees the device can accommodate the developing demands of customer service operations. Destiny enhancements should encompass the combination of gadget mastering for computerized ticket categorization and sentiment evaluation, which might further enhance price tag management and prioritization. additionally, incorporating AI-pushed chatbots should offer customers on the spot responses, enhancing the general help experience.

In end, this system illustrates how automation can remodel customer support tactics, providing a robust and scalable answer that can be tailor-made to in shape the needs of diverse companies.

REFERENCES :

[1] *Ali, S., & Raza, M. (2021).* "Automation in Customer Service Systems: A Comprehensive Review." *International Journal of Computer Science and Information Security*, 19(7), 75-85.

[2] *Ameen, A., & Khan, M. (2020).* "Artificial Intelligence in Customer Service: A Survey." *Journal of Artificial Intelligence Research*, 49(1), 102-115.

[3] *Chung, W. (2019).* "Trends in Ticketing Systems for Customer Support." *Journal of Technology in Customer Service*, 21(2), 45-60.

[4] *Bashir, S., & Iqbal, M. (2022).* "The Role of Natural Language Processing in Customer Support Automation." *Journal of AI and Computational Linguistics*, 27(5), 130-145.

[5] *Dey, S., & Ray, A. (2018).* "Django for Building Scalable and Secure Web Applications." *Proceedings of the International Conference on Web Technologies*, 10(1), 85-94.

[6] *Singh, R., & Kumar, R. (2020).* "MySQL: A Reliable Data Management System for Modern Web Applications." *International Journal of Computer Engineering*, 12(6), 243-258.

[7] *Kim, Y., & Lee, H. (2021).* "Customer Satisfaction through Automation: Analyzing the Impact of Ticketing Systems." *Journal of Service Science and Management*, 13(3), 222-239.
[8] *Mishra, D., & Sharma, P. (2021).* "Ticket Classification in Customer Support Systems using Machine Learning." *Journal of Computational Intelligence in Business*, 14(3), 100-115.
[9] *Sahu, M., & Ghosh, S. (2020).* "AI-Powered Chat-bots in Customer Support: A Future Trend." *International Journal of AI in Business*, 29(4), 400-414.
[10] *Jain, R., & Gupta, V. (2021).* "Enhancing Customer Support through Automation and AI." *Journal of Business and Technology*, 18(2), 120-135.
[11] *Sharma, A., & Agarwal, S. (2019).* "The Evolution of Customer Support Systems: From Manual to Automated." *Global Journal of Business Management*, 16(4), 189-203.
[12] *Patel, V., & Shah, A. (2020).* "Data Handling in Automated Systems: Using MySQL for Performance and Scalability." *Journal of Database Systems*, 19(2), 55-70.
[13] *Hassan, S., & Lee, K. (2022).* "Integrating NLP with Customer Support: A Detailed Approach." *AI and Business Journal*, 28(6), 78-90.
[14] *Sharma, P., & Mehta, A. (2020).* "Scalable Web Applications using Django Framework." *International Conference on Web Development*, 11(2), 66-80.
[15] *Ali, Z., & Tan, L. (2021).* "The Role of Automation in Reducing Customer Service Costs." *Business Process Management Journal*, 27(5), 400-415.
[16] *Rani, S., & Tiwari, P. (2019).* "Challenges in Implementing Automated Customer Support Systems." *International Journal of Business Automation*, 11(3), 82-96.
[17] *Kumar, V., & Sharma, R. (2020).* "Machine Learning in Automating Customer Queries: A Case Study." *Journal of Machine Learning in Business Applications*, 15(2), 110-126.
[18] *Bharadwaj, R., & Singh, H. (2020).* "AI and Automation in Customer Support: Best Practices." *Journal of AI and Customer Relationship Management*, 21(1), 12-28.
[19] *Lo, J., & Chen, M. (2021).* "AI for Customer Service Automation: Trends and Future Directions." *AI Review Journal*, 30(6), 500-510.
[20] *Mason, D., & Clark, T. (2022).* "Evolving Customer Service Platforms with Intelligent Ticketing Systems." *Journal of Emerging Technologies in Business*, 14(2), 95-105.
[21] *Patel, S., & Kumar, R. (2021).* "Leveraging Automation for Customer Service Excellence." *Global Customer Support Review*, 8(4), 67-80.
[22] *Feng, J., & Liu, S. (2019).* "Customer Support and AI Chatbots: A Study on Efficiency and Engagement." *Journal of AI in Business and Customer Support*, 17(1), 45-59.
[23] *Smith, R., & Williams, H. (2020).* "Redefining Customer Support with Automation and Machine Learning." *Journal of AI and Technology Solutions*, 22(3), 140-150.
[24] *Ali, M., & Shah, I. (2022).* "The Role of NLP in Automating Customer Queries." *Journal of Natural Language Processing and Automation*, 25(4), 210-225.
[25] *Gupta, S., & Mishra, A. (2021).* "Optimizing Ticket Management with AI: An Integrated Approach." *Journal of Intelligent Systems in Business*, 18(3), 400-420.
[26] *Mehta, N., & Sharma, V. (2019).* "Enhancing Efficiency in Customer Support Systems using Machine Learning." *International Journal of Business Intelligence*, 19(2), 98-112.