

# Pager Duty - Intelligent Incident Management

<sup>1</sup>Supriya Singh

UG Scholar(B.Tech)  
Indira Gandhi Delhi Technical  
University For Women, Delhi  
[supriya065btece22@igdtuw.ac.in](mailto:supriya065btece22@igdtuw.ac.in)

<sup>2</sup>Sonika Chamola

UG Scholar(B.Tech)  
Indira Gandhi Delhi Technical  
University For Women, Delhi  
[sonikachamola@gmail.com](mailto:sonikachamola@gmail.com)

<sup>3</sup>Vidushi Bharadwaj

UG Scholar(B.Tech)  
Indira Gandhi Delhi Technical  
University For Women, Delhi  
[vidushi075btece22@igdtuw.ac.in](mailto:vidushi075btece22@igdtuw.ac.in)

<sup>4</sup>Shreya Koli

UG Scholar (B.Tech)  
Indira Gandhi Delhi Technical  
University For Women, Delhi  
[shrevakoli2519@gmail.com](mailto:shrevakoli2519@gmail.com)

<sup>5</sup>M. Ejaz Aslam Lodhi

Electronics and Communication Department  
Indira Gandhi Delhi Technical  
University For Women, Delhi  
[ejaz@igdtuw.ac.in](mailto:ejaz@igdtuw.ac.in)

**Abstract**—This research paper presents the design of an intelligent and minimal incident management system which is implemented for small businesses, startups, and personal projects of teams. Managing technical issues when dealing with incidents in the real world can be messy and stressful, which can result in lost clients, unsuccessful and broken trust. For such teams juggling multiple responsibilities, existing systems, such as PagerDuty, are sometimes too complicated, costly, and challenging to implement. This paper proposes a system which addresses this gap by providing an easy-to-use incident management system. Features like incident logging, SMS alerts, and a simple dashboard; help in solving problems quickly and maintaining consistency. By focusing on a less complex model and flexibility the system depicts how small-scale projects can improve response time, reliability, and consistency. It can evolve as needs grow or technology changes.

**Index Terms**—Incident Management, SMS Alerts, Categorisation, Prioritisation, Dashboard.

## I. INTRODUCTION

Small businesses, startups, and personal projects increasingly depend on online platforms in today's digital age for vital operations, interaction with clients, and earning income Ref. [1]. However, even minor technical difficulties such as interruptions in the system, transaction failures, or feature glitches may lead to severe issues Ref. [2]. These include losing customers, decreased revenue, and a breakdown of trust. Ref. [3] builds an economic model quantifying the cost of IT-service downtime for firms. It explicitly analyzes how downtime (service outages) imposes financial cost on firms. Though there exists multiple incident handling systems, they are complicated, expensive, and demand settings that are not suitable for smaller environments Ref. [4], Ref. [5], Ref. [6]. This highlights the importance of an incident management system that is simple to operate, user-friendly and recognizes the unique requirements and constraints of small organizations and individual developers Ref. [7]. Effective incident management requires detecting problems early, reporting them without delay, arranging them clearly, and addressing them efficiently to keep disruptions to a minimum Ref. [8]. A well-planned system that is easy to navigate, sends alerts when needed, provides frequent updates on ongoing tasks, and records all activity can help small-scale teams act quickly and keep operations clear and transparent Ref. [9]. Well-defined procedures and accessible records help teams remain accountable and make it easier to review past incidents and learn from them Ref. [10]. A straightforward and affordable system allows small teams with limited IT resources to stay organized, keep workflows running smoothly, meet user expectations, and avoid the high costs of enterprise-level solutions Ref. [11]. By providing a centralized view of incidents, teams can more easily identify recurring problems and address their root causes. Simple tracking methods reduce confusion during critical periods and ensure that no incident is overlooked. Ultimately, these features help small teams operate effectively, maintain customer trust, and expand with confidence in spite of inadequate technical infrastructure Ref. [12].

This research explains the design and creation of an incident management system using Node.js, Express, SQLite, and Nodemailer, which can be accessed through a web interface and REST API. The system is built to be simple and practical, offering features such as manual incident entry, email notifications, status tracking, and a clear dashboard for monitoring incidents, enabling teams to handle incidents effectively without unnecessary complexity or expense. By providing a solution suited to smaller-scale needs, it helps close gaps in incident management tools and supports the smooth functioning of emerging digital ventures. The system also enables teams to prioritize incidents by severity, making sure that the most critical issues are dealt with promptly Ref. [13]. With incidents organized clearly and accessibly, important information is always easy to find, allowing for timely action and informed oversight. This combination of straightforwardness and clarity helps users stay accountable, track patterns, and make sound decisions during and after incidents. Overall, the system shows how lightweight tools can help teams maintain operational reliability and continuity, even when technical resources are limited Ref. [14].

## II. SYSTEM ARCHITECTURE AND DESIGN

### A. Component Architecture

The Basic Incident Management (BIM) dashboard is a multi-layered architecture tackling a variety of concerns. There are separate layers for Data storage, Data Management, Data Analysis and user interaction. The system is designed with a focus on simplicity, modularity, and efficiency. This structure allows each component to function independently and maintain a seamless workflow between them. The [incidents.db](#) file (github file) stored the SQLite Database containing records of all incidents. The usage of SQLite makes it simpler, reliable and data storage easy for a small scale dashboard, in the future we can move to a bigger database making it more scalable(future recommendations). All separate files contribute to the simplicity and modularity, promoting reusability.

### B. Technology Stack Selection

The technology stack was chosen based on criteria of rapid development, zero-configuration complexity, and low operational cost, directly supporting the objective of affordability and ease of implementation.

Table 1 Technology Stack And Rationale

Component	Chosen Technology	Justification
Backend Framework	Node.js + Express	Lightweight, efficient for simple API routing and I/O tasks.
Database	SQLite	Serverless, zero-configuration embedded database, affordable
SMS Alerts	Twilio SMS	Simple integration, secure, critical communication.
Frontend	HTML + CSS	Prioritizes functionality and speed.
Data Analysis/ML	Python + pandas / scikit-learn / mlxtend	Rich ecosystem for data processing, statistical analysis.
Visualisation Libraries	Matplotlib + Seaborn + NetworkX	Powerful plotting libraries for generating charts, scatter plots, and network graphs to visualize data insights clearly.
Task Scheduling	GitHub Actions	Automate recurring
		tasks like data updates, backups and provide version control.

C. Layered Architecture

The presentation layer is implemented as a dashboard built using HTML and CSS displaying the incident reports, columns to be filled along with an optional SMS support. It is beneficial as it separates the business logic from the user interface. Due to the usage of HTML and CSS there is a great scope for an extendable and richer frameworks. Because the system uses a web-based dashboard (HTML/CSS), it becomes extendable and user-friendly, allowing teams to view, track and manage incidents easily Ref. [15]. It allows users to create and view incidents through a dashboard interface. The dashboard communicates with the backend ensuring real-time synchronization of incident data. The simplicity of the design allows for even a person not well-equipped with technical knowledge to understand.

The application layer, developed using Node.js and Express, implements all core logics such as incident creation, deletion and works in accordance with third party integrations. Twilio SMS services are used to alert the user once their submitted data is entered into the database. This allows for a more dynamic and transparent exchange of data. The creation of this layer has been done in an independent manner, meaning all the various components are independently working together.

The data access layer employs SQLite, a lightweight and serverless relational database system. It contains a single table, which stores incident ID, title, description, severity, status, type of website, any additional tags, frequency of occurrence of the problem. This simple schema provides efficient handling and minimizes complex structures, aligning with the project’s goal of creating an affordable and easily deployable solution.

The data layer also comprises SQLite, it is a relational database. It is simple, serverless and zero-configuration. The most apt tech stack for a small-to-medium dataset.

Analysis Layer is a simple Machine learning layer that analyses patterns among the data points collected through the dashboard. Cleaning and transforming of data into insights is facilitated by this layer. Our aim is to be able to come up with insights for solution-making companies, performing the analysis for them to be able to understand where the modern websites fail.

External SMS alerts were integrated by creating a twilio account, generating codes and credentials. This layer is the proof that our dashboard is compatible with the existing technologies Ref. [16].

1) System Workflow

The Basic Incident Management system operates through a smooth, stepwise flow that ensures incidents are captured, tracked, and addressed efficiently. When a user uploads a report via the frontend, incident details containing all the parameters are sent to the backend (Node.js + Express), all records are saved in the SQLite layer. Incidents are queried, updated, prioritized. SMS alerts are sent forward by Twilio.

On the other layer, the Python ML module loads incident data from SQLite. Data is preprocessed, association rule mining discovers recurring patterns among incidents. Visualisation of this data allows for a more detailed explanation.

Final insights are derived on the basis of the many ML models, to generate solid conclusions that can be treated as a market analysis to understand system failures and generate system specific-solutions.

ID	Title	Category	Priority	Status	Phone	Website Type	Incident Frequency	Service Affected	Root Cause Category	Tags
1	Slow Homepage Load	General	P1	Open	+917555672345	Ecommerce	one-time	login	frontend	slow loading
2	Checkout Payment Failure	Payments	P1	Open	+017866734567	Tour Booking website	intermittent	checkout	third-party	urgent
3	API Timeout with Third-Party	Network	P1	Open	+915877654312	Ecommerce	continuous	API	third-party	bug
4	Data Sync Delays	General	P1	Open	+912345443234	Ecommerce	one-time	other	backend	operations issue
5	Login Authentication Error	Authentication	P1	Open	+917866543456	Quick Commerce	one-time	login	frontend	login blocked
6	Database Lock Contention	Database	P1	Open	+916577890765	Tour Booking website	one-time	search	database	slow load
7	Slow page load on homepage	General	P1	Open	+919821334561	Ecommerce	one-time	other	frontend	

Fig. 1 Online incident reporting form - Basic Management System

Fig. 2 Incident records display showing current open issues

D. Design Considerations

The design is kept simple and the architecture is deliberately left open-ended to ensure that it can easily be made scalable in the future. The frontend is an interactive user interface, made using HTML and CSS. It is made in a way that even someone who isn’t familiar with the new automated technologies, can easily get well acquainted with the design. The use of Node.js and SQLite is done so that the system remains lightweight and modular. They facilitate easy installation and handling of the incidents Ref. [17]. The integration of SMS alerts gives a very low latency and it bridges the communication gap with the users as it notifies them when their issue is raised in real-time. The Machine Learning implementation offers great insights into the incident data, giving analysis of the different incident failures and the reasons behind them along with the categorisations and the frequencies of their occurrences. This can help in the market analysis.

III. AUTOMATED INCIDENT CATEGORISATION & PRIORITISATION

The incident management system also deals with the categorisation and prioritisation of the incident data based on several categories and priorities. The system automates these two processes to save the time it would otherwise take for a human to manually categorise. The prioritisation makes sure that the most critical incidents are dealt with the soonest. This improves the efficiency, accuracy and causes less dissatisfaction in the users.

### A. Incident Categorisation

It is fundamental to properly categorize the incident data. The proper classification early on essentially helps teams that are starting off or have very little historical data to derive categories from. This helps in the study of this data and helps teams deal with the right kind of issues the right way. The system works with a rule-based classification method which then scans the incident database for details and keywords to classify the incident into one of the few predefined categories (Database, Network, Authentication, Payments, and more). This rule-based classifier helps in categorizing data consistently right from the start. The system streamlines how the issues are understood and resolved. The team can effectively grasp the nature of the problem, and understand the kind of issue raised and what team should the issue be tackled by is then chosen accordingly. The early startups can focus their efforts efficiently and the issues are raised to the right teams without delay.

### B. Incident Prioritisation

Once the incidents are categorized, these are further prioritized enabling the critical issues to get the fastest attention. The system tries to balance urgency and impact by prioritizing incidents based on the columns from the incident database that store factors like severity, frequency and the repetitions of the issues. It also simultaneously checks for keywords that display urgency like “timeout”, “outage” or “critical”.

Further some tags in the description can also indicate how critical a particular issue is. Some of these tags include “urgent”, “panic”, etc. These tags also escalate priority issues.

Priority indicators ranging from P0(critical) to P2(normal) are assigned to incidents based on these above keywords and factors that help decide what incident needs to be tackled first and foremost for the efficient and smooth working of the model. The end goal is to ensure that together with classification, incident prioritisation reduces the manual workload, ensures consistency and mitigates delays to enable high quality service and a good system in place for managing issues on time and making system and services reliable for the users.

### C. Benefits

The incident categorisation and prioritisation comes with several different benefits:

1. Teams have to spend less time manually analyzing the incoming data due to the automation of this classification system.
2. The response cycle becomes faster and website crashes or system failures are prevented by focussing on the most critical incident failures which can heavily impact the entire system.
3. Enables deeper data analytics which can support the systematic handling of problems.
4. Mitigates human errors, reduces cognitive load and gives a reliable process to keep the functioning of the small scale projects in check.

### D. Integration with the Proposed System

The two modules namely: Incident categorisation and Incident prioritisation, are tightly integrated with the proposed system into the incident creation workflow. In real-time when an incident is created and submitted, the backend simultaneously analyzes this incident data and classifies it and takes into account all details including affected services, tags, incident recurrence(frequency) and the cause. Timely insights are made available for the responders and since this data processing doesn't require manual input, it is faster and more accurate. The integration of these two modules creates an efficient and effective continuous feedback loop system which improves system accuracy and response.

### E. Future Potential

Our current system uses a rule-based classification approach and even though it is highly effective for a small number of incident entries, the future potential of integrating machine learning models for the categorisation and prioritisation tasks can only broaden the implementation of this model on large-scale. With the accumulation of more labeled incident data, machine learning models can improve the accuracy and can perform these tasks much faster. The influence of ML in the model will help predict rising trends or impending outages much before they actually occur and can help teams tackle the issues beforehand. This will enable the system to learn from its own experience and become smarter.

## IV. NOTIFICATION AND ALERTING SUBSYSTEM

The incident management system can be considered as just an incident tracking system. This underlines a very passive tracking approach. Using Notification and Alerting Subsystems powered by Twilio's SMS integration enables the switching of this passive approach to a more interactive and proactive approach. The usual method of getting updates on email can sometimes delay the process of incident handling where critical incidents are to be resolved. Sending SMS text messages directly on the mobile phone devices of key stakeholders prevents the chances of missing an alert or the delay in its response.

Twilio has a cloud communication platform which is highly secure and dependable. It systematically sends SMS text messages right after an incident is logged and has a very straightforward approach. It follows a direct line of communication. This direct line of communication cuts through the noise of information overload and ignores the slower channels of communication and hence significantly decreases the latency in response time. Since teams get timely insights, they can take action and mitigate impacts before the issues even escalate. So the alerts received by the responders are clear and instant.

The global infrastructure of Twilio results in the wide coverage area of communication wherein the deliverability of messages is high. Twilio also tracks and records all messages being transmitted and maintains a log of that so that the record of the time and the member who received the message is also made available to the startup or small team. This enables transparency and helps

achieve high accuracy and imparts compliance during the incident failures. It integrates the value of accountability within the team.

The Notification and SMS Alert Subsystem using Twilio makes the model dependable, reliable and enhances situational awareness continuously.

**V. RESULTS AND EVALUATION**

The evaluation of the Basic Incident Management System focused on how well it performed under different conditions, for this we evaluate the following-antecedents, consequences, confidence, support and lift. The starting conditions for the association model. “Tag:deployment”, “Tag:rollback”, “Tag:partial” are all tags that show the initial triggers for association. “Tag:rollback”, “Tag:partial” and “Tag:outage” are also tags that show the consequences to the antecedents. They show outcomes with association in the form of tags. The probability that the consequence occurs when the antecedent occurs is the probability. The frequency of combination of the antecedent and the consequence is the Support. Our final parameter, lift, measures the strength of association, values above 1 indicate strong relationships. The graphs plotted for the above parameters are shown below.

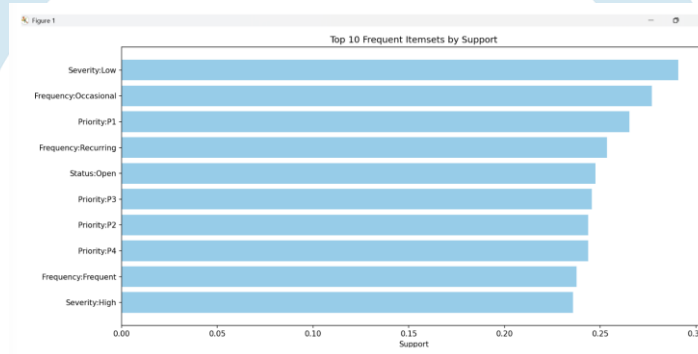


Fig. 3 Analysis of Frequent Itemsets Showing Incident Severity, Priority, and Frequency

The x-axis shows the proportion of dataset that each itemset shows, a higher value represents a higher frequency of the data. The y-axis includes a list of labels representing severity, priority and frequency. The key conclusions are as follows:

1. “Severity:low” tag has the highest support, which means most of the problems reported are of low severity. If given an option between permanent but slowly progressing solution or a temporary quick solution, they would go for the first option.
2. There is a lot of diversity in Item types. Categories “Status:Open” and “Severity : high” are included among the top 10 but appear less frequently than :severity:low”. By taking into consideration all these parameters we perform the categorisation and prioritisation.

To optimize incident response, this graph helps us in understanding the dominating datasets. Understanding how much a parameter affects the final output helps us in establishing urgencies and loopholes in the web-solutions that are present out there.

Percentage Distribution of Website Types Reporting Issues

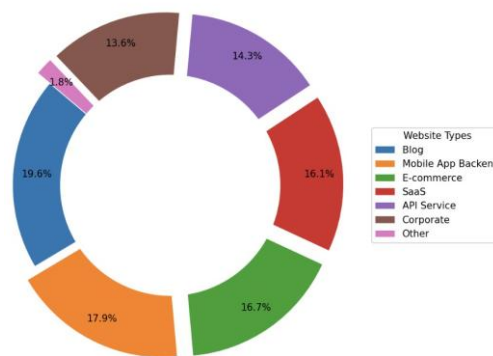


Fig. 4 Percentage distribution of website types reporting issues

The above pie-chart shows the various categories of websites/dashboards that registered an issue with their functionality. The dominant category with 19.6% happen to be Blogs, other significant contributors are Mobile App backend (17.9%), E-commerce(16.7%), and API services (14.3%). This represents that a different variety of platforms experience this turbulence of shutdown and issues but those directly related to the client interaction or service delivery, like mobile apps, e-commerce and SaaS are the majority of the representation in such a case.

Another minimal category is the corporate sites contributing to 13.6%, this indicates that even less dynamic and interactive sites experience these issues and breakdown. Hence expanding the market to solution-creating partners as small to medium websites comprise a big portion of our carbon footprints Ref. [18].

Testing involved both the web interface and the REST API to work on different scenarios. Each incident created during these tests appeared in the SQLite database, one thing to keep in mind is when working on the DB, for all changes made the database must be initialised again from the root node. This allows for changes in the SQLite Database to appear on the frontend. Failing to do so

might lead to no data storage on the frontend, changes will be recorded only in the Database.

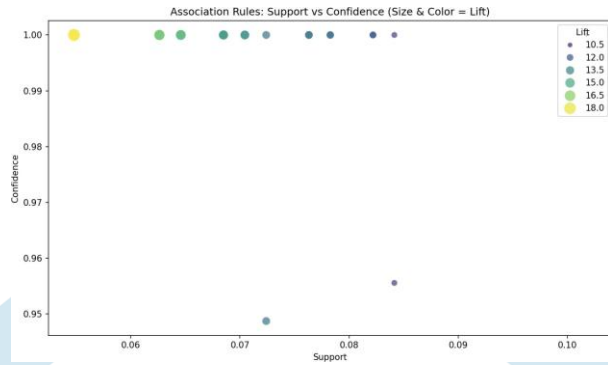


Fig. 5 Relationship between lift and support

SMS notifications are sent just as a user enters an incident in the database. This SMS facility has been enabled in association with Twilio SMS services.

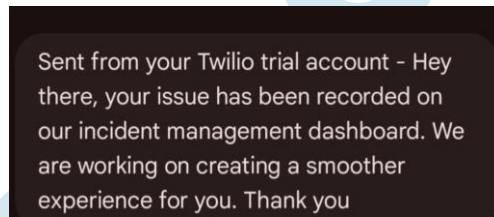


Fig. 6 Screenshot received when incident is registered

A number verification is needed on the Twilio website, once the user verifies their number. Their number will start receiving SMS alerts every time an incident is entered from their mobile number. All credentials can be accessed using the official Twilio website.

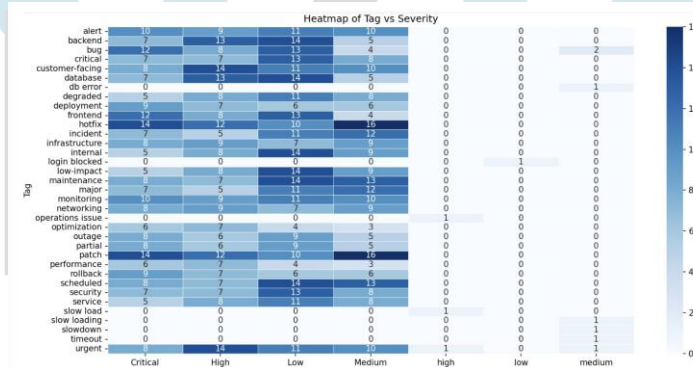


Fig. 7 Heatmap of tag vs severity

Performance tests during back to back submissions showed that the Express.js server remained responsive and there were no crashes or errors when multiple entries were inserted together. This shows that the chosen stack Node.js, Express, and SQLite is suitable for small to medium-scale setups. A large part of the web-tech world, this allows for newer markets to open up for companies that make solutions.

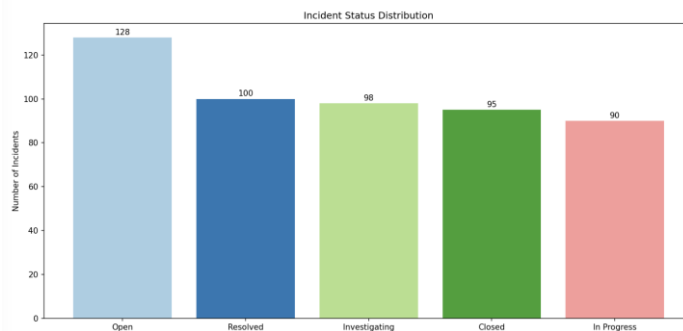


Fig. 8 Bar chart of incident status

Some limitations too were noted. The database being used currently is suitable for a small scale enterprise and reporting. To make it scalable and viable for medium to large scale enterprises, the database needs to be expanded. The SMS alert feature is currently functioning on the free trial, and it requires the user to verify their number to be able to receive SMS alerts. We need to upgrade to a paid package to be able to reach out to a wider audience. There are no specified accesses for the dashboard manager and

dashboard user, all data being entered can be seen by anyone who opens the dashboard, to maintain privacy and security a two-way system needs to be developed.

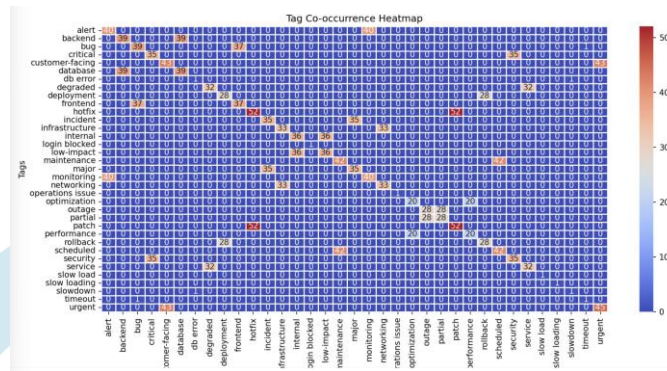


Fig. 9 Co-occurrence heatmap

Overall, the system reacted quickly to incidents. It handled data smoothly and was easy to set up. Maintenance was minimal, which is more convenient for smaller teams. Its open-source nature and lightweight setup make it a practical, low-cost option for academic projects, startups, or small businesses that need reliable incident management at a low cost.

## VI. ML POTENTIAL

The Basic Incident Management System currently operates on a single ML model framework, the Association rule mining, it identifies patterns in data and computes if-else categories (“if antecedent, then consequent”) it derives relationships among the various patterns. We have used the apriori algorithm, which works according to the following rules.

1. It identifies frequent itemsets and combinations.
2. Generates candidate rules from those items.
3. Evaluates each rule using metrics such as support, confidence and list.

However the conclusions can be made much more stronger and with higher accuracy and precision, once we implement other models too.

Server Downtime can be prevented by applying Naïve Bayes, SVM, or Logistic Regression. We are currently asking users to add optional tags to their incident, this facilitates and makes extraction easier. We also categorise on the basis of the data received with the help of our backend application layer.

The common applications of this are for the market base analysis to find products that are frequently purchased together. In our cause it helps us in understanding what type of website faces which issue. For example Ecommerce websites face higher traffic surges during festive seasons. Identifying recurring relationships and recommending actions on the basis of these patterns. Implementing machine learning requires larger datasets, regular retraining, and integration with frameworks like TensorFlow or Scikit-learn. When one moves towards developing a more scalable dashboard, ML can be extensively used and incorporated for strengthening our conclusions and generating meaningful inferences.

## VII. FUTURE WORK

In the next stages of development, the system will add features that make it more secure, scalable, and helpful for all types of users. One such expansion is of the Dataset, the current model is built on 509 data entry points. To be able to create conclusions that can be worked upon we need more data points. We have implemented the association model, and in the future the system should aim at implementing various advanced algorithms like the FP-Growth, Eclat and temporal association. This will improve efficiency and scalability.

Integration of the association rule mining model with predictive analysis like classification, time series forecasting and many more. Automating responses in different application areas can also prove to be helpful.

Expanding our applications in other domains like healthcare, social media, finance and quick commerce can lead to a wider variety of data and provide actionable insights.

Another one of the main improvements will be the addition of user accounts and logins for the dashboard. This means that each person who wants to view or manage incidents will need to sign in to the system, making sure that only people with permission can access important information. Along with this, users will have different roles, such as admins or responders. Admins can control every part of the system and manage users, while responders focus on handling and updating incidents. This setup not only makes the system safer but also means jobs are clear, and everyone knows what they are supposed to do. As the system grows, these features will be important for building trust and making sure information stays private and secure.

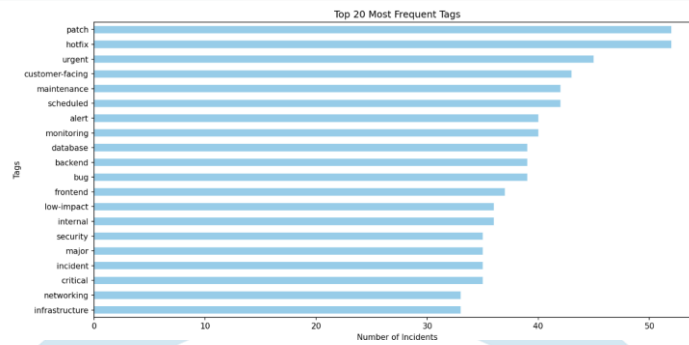


Fig. 10 Top 20 Most Frequent Tags

Finally, the system will offer simple analytics and reports that show trends and important stats about incidents in the form of a document presented to the user also with the integration of Twilio. For example, users will be able to see how often problems happen, how quickly they are resolved, and if certain issues keep coming back. These insights will help companies investing in solutions and creating them to be able to find weak spots, improve their workflows, and sell this to the companies in need. By adding these features, the system will move from being a basic tool for reporting incidents to a more complete management platform. One that allows people to report when something goes wrong and also allows solution oriented companies to create web solutions that are specifically catered to these websites.

### VIII. CONCLUSION

The Basic Incident Management System was developed to provide a straightforward and simple way to report, track, and manage incidents. Its interactive and simple design, built using Node.js, Express, and SQLite, ensures quick setup, low maintenance, and efficient performance. The system simplifies the entire process, from creating an incident to notifying users and updating its status, making it well-suited for small teams or businesses that need a functional alerting solution without added complexity.

```

apriori_results_conclu... 1 | data
1 | conclusion_number, antecedents, consequents, confidence, support, lift
2 | 1, Tag: deployment, Tag: rollback, 100.0%, 5.3%, 18.18x
3 | 2, Tag: rollback, Tag: deployment, 100.0%, 5.3%, 18.18x
4 | 3, Tag: outage, Tag: partial, 100.0%, 5.3%, 18.18x
5 | 4, Tag: partial, Tag: outage, 100.0%, 5.3%, 18.18x
6 | 5, Tag: degraded, Tag: service, 100.0%, 6.3%, 15.91x
7 |

```

The above image shows the conclusions derived on the basis of the 509 data points collected.

1. The above analysis identified that “severity:low” and “frequency:occasional” are the most common categories. which can be inferred as, solutions need to be created that are permanent and can handle the occasional spike. A subscription model of these solutions can be run to offer it only during the occasional spikes.
2. The majority that reported issues were Blogs, Mobile backend services, E-commerce, SaaS and API platforms.

By incorporating the future scope into our system. Making it more robust and scalable, we can create the one central hub for incident reporting, analysing and taking action.

In essence, the project achieves its goal of delivering an accessible and efficient incident management tool and at the same time is still in its growth stage. Its modular structure and open design leave room for further development, allowing it to evolve into a smarter, more collaborative platform in the future.

### IX. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Mr. M. Ejaz Aslam Lodhi, Electronics and Communication Department, Indira Gandhi Delhi Technical University for Women, Delhi, for his guidance, constant support, and valuable feedback throughout the course of this research. His suggestions and encouragement played an important role in shaping the direction of this work.

The authors also wish to thank Indira Gandhi Delhi Technical University for Women for providing the academic environment and resources that supported this research.

Finally, the authors acknowledge the efforts and dedication of all team members who contributed to the development of the DebugNow platform and the preparation of this paper.

### REFERENCES

- [1] M. A. Hossain, N. Jahan, A. Masud, M. N. Nabi, M. S. Hossain, and S. Ahmed, “Dynamic effect of critical success factors of SMEs on entrepreneurial performance via e-commerce performance,” *J. High Technol. Manag. Res.*, vol. 35,

no. 2, p. 100515, 2024.

- [2] J. H. Anthony, W. Choi, and S. Grabski, "Market reaction to e-commerce impairments evidenced by website outages," *Int. J. Account. Inform. Syst.*, vol. 7, no. 2, pp. 60–78, Jun. 2006.
- [3] S. S. Wang and U. Franke, "Enterprise IT service downtime cost and risk transfer in a supply chain," *Operations Management Research*, vol. 13, pp. 94–108, 2020.
- [4] T. Tam, A. Rao and J. Hall, "The good, the bad and the missing: A narrative review of cyber-security implications for Australian small businesses," *Computers & Security*, vol. 109, art. no. 102385, 2021.
- [5] M. Benz and D. Chatterjee, "Calculated risk? A cybersecurity evaluation tool for SMEs," *Bus. Horizons*, vol. 63, no. 4, pp. 531–540, 2020.
- [6] J. Manzoor, A. Waleed, A. F. Jamali and A. Masood, "Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs," *PLoS ONE*, vol. 19, no. 3, p. e0301183, 2024.
- [7] S. Chaudhary, V. Gkioulos, and D. Goodman, "Cybersecurity Awareness for Small and Medium-Sized Enterprises (SMEs): Availability and Scope of Free and Inexpensive Awareness Resources," in *Computer Security. ESORICS 2022 International Workshops*, Lecture Notes in Computer Science, vol. 13785, S. Katsikas, F. Cuppens, C. Kalloniatis, J. Mylopoulos, F. Pallas, J. Pohle, and others, Eds. Cham: Springer, 2023, pp. 97–115.
- [8] Q. Yu, N. Zhao, M. Li, Z. Li, H. Wang, W. Zhang, K. Sui, and D. Pei, "A survey on intelligent management of alerts and incidents in IT services," *J. Netw. Comput. Appl.*, vol. 224, art. no. 103842, 2024.
- [9] K. Meléndez, A. Dávila, and M. Pessoa, "Information technology service management models applied to medium and small organizations: A systematic literature review," *Comput. Stand. Interfaces*, vol. 47, pp. 120–127, 2016.
- [10] M. Shetty, C. Bansal, S. Kumar, N. Rao, and N. Nagappan, "SoftNER: Mining knowledge graphs from cloud incidents," *Empirical Software Engineering*, vol. 27, art. no. 93, 2022.
- [11] D. P. Feversani, V. De Castro, E. Marcos, M. G. Piattini, and M. L. Martín-Peña, "Towards a lightweight framework for service management evaluation in SMEs," *Inf. Syst. e-Bus. Manage.*, vol. 21, pp. 81–122, 2023.
- [12] A. Peralta, J. A. Olivas, F. P. Romero, and P. Navarro-Illana, "Intelligent Incident Management Leveraging Artificial Intelligence, Knowledge Engineering, and Mathematical Models in Enterprise Operations," *Mathematics*, vol. 13, no. 7, Art. no. 1055, 2025.
- [13] R. Gupta, K. Prasad, and M. Mohania, "Automating ITSM incident management process," in *Proc. IEEE Int. Conf. Autonomic Computing (ICAC)*, Chicago, IL, USA, Jun. 2008, pp. 141–150.
- [14] D. MacLean and R. Titah, "Implementation and impacts of IT Service Management in the IT function," *Int. J. Inf. Manage.*, vol. 70, art. no. 102628, Jun. 2023..
- [15] M. Holjevac and T. Jakopec, "Web application dashboards as a tool for data visualization and enrichment," in *2020 43rd Int. Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2020, pp. 1740–1745.
- [16] C. A. Figueroa, T. C. Luo, A. Jacobo, A. Munoz, M. Manuel, D. Chan, J. Canny, and A. Aguilera, "Conversational Physical Activity Coaches for Spanish and English Speaking Women: A User Design Study," *Front. Digit. Health*, vol. 3, art. no. 747153, 2021.
- [17] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, Nov.–Dec. 2010.
- [18] O. Philippot, A. Anglade, and T. Leboucq, "Characterization of the energy consumption of websites: Impact of website implementation on resource consumption," in *Proc. 2014 Int. Conf. ICT for Sustainability (ICT4S 2014)*, Paris–Île-de-France, France, Aug. 2014, pp. 171–178.