

DIGITAL CAMPUS QUERY ACQUISITION SYSTEM

Gowthama Kurinji Veandan R M, Gokul J

¹Student, Computer science and engineering, Bannari Amman Institute of Technology, Erode, India

²Student, Computer science and engineering, Bannari Amman Institute of Technology, Erode, India

ABSTRACT

Digital Campus Query Acquisition System is an essential tool to provide the solution and track the queries of different problems gathered from employees in digital campus and getting a reply from the respective teams. The queries mean to be like the employee access issues, DevOps issues, data issues, laptop issues and other hardware & software issues. There will be a separate website for the IDP digital campus, and all the employees will have the account in that website. The website contains the query acquisition service where employee can be raising their queries. The queries will get assigned to the appropriate team handling that service. The queries can be assigned to different teams and will get tracked to closure. The team handling that queries will have the facility to communicate with the employee via Microsoft teams through the system. And the queries submitted by the employee will be stored in the relational database service in AWS using PostgreSQL. Status of the issue will be tracked through mail using SMTP service. Overall, the digital campus query acquisition system is to gather, track, respond and to rectify the queries and issues for the digital campus employees.

Keyword: - Query Acquisition, AWS, SMTP, PostgreSQL.

1. INTRODUCTION

In Digital Campus, there has been a service desk team which includes DevOps team, CloudOps team, Database team, System and network administrators etc., And the employees working in digital campus having dependencies with the service desk team on various issues and queries which are more related to several access requirements, system issues, installation queries and dependencies and technical dependencies too. As for these dependencies, Employees must go to the service desk team for the solution and there is no proper appropriate way to get the queries tracked and the rectifying the issues process is quite harder in this current process. And the issue/query is later can't be verified by whom it was resolved, and it can't be tracked. Hence to overcome the current process, Digital Campus Query Acquisition System is implemented to overcome the above uncertainty.

This system usually allows employees to submit questions, concerns or requests for assistance through a various channel, including email, web forms or chatbots. System captures all queries and stores them in a centralized database, allowing organization to track and manage the status of each inquiry from submission to difference. By automating the query search process, the system reduces the risk of lost or unanswered questions helps employees receive timely responds and provides management with valuable insights about the employee concerns and trends. Finally, a query search system can help organizations improve communication, increase productivity and empower employee's satisfaction.

A query acquisition system is an essential tool for any organization is trying to streamline the process of handling employee inquiries. This project uses performance of several cutting-edge technologies, including AWS, Python, Terraform and PostgreSQL to create a highly efficient and scalable system. Using AWS, the system can easily handle large volumes of queries maintaining high availability and reliability. Python is used for development application logic while terraform is used to automate deployment and basic infrastructure management. PostgreSQL provides a robust and flexible database for storing and querying data captured by the system. Together, these technologies create a powerful solution that enables organizations to effectively collect and manage employee inquiries and ensure that all queries are tracked until resolved.

2. LITERATURE SURVEY

Bokanyi, Krahmer and Wubben (2020) found that chatbots can effectively handle routine queries, allowing human agents to focus on more complex problems. A considerable body of literature relates to this query management and query retrieval systems. Several studies have been reviewed the effectiveness of different approaches to managing employee enquiries, including the use of automated systems and chatbots [1].

Kaur, Rana, and Sharma (2017) designed a query management system that used a combination of rule-based and machine learning techniques for categorization and direct inquiries to the appropriate department. It was found that the system significantly improves query resolution times and reduce the workload of human agents [2].

Omogbe, Onibere, and Oyewole (2019) examined the use of questioning management systems in the Nigerian banking industry, finding that such systems can significantly improve customer satisfaction and reduce operating costs. Several studies were also reviewed. the use of query management systems in specific industries [3]

Hassan, Razzaq, and Iqbal (2019) found that AWS-based query management systems could significantly reduce and reduce query resolution time operating costs in call centers. a growing body of research highlighting the effectiveness of AWS in development and implementing query management and query retrieval systems [4].

Ramakrishnan and Srinivasan (2018) investigated the use survey of health care procurement systems and found that such systems can significantly improve patient outcomes and reduce operating costs. several studies have been conducted explored the use of query retrieval systems in specific industries [5].

Overall, the literature suggests that query management and query retrieval systems can be highly effective improving communication and reducing the burden in organizations. Use advanced technologies such as chatbots may continue further increase the efficiency of these systems. And the Major Finding from the literature leads us to find the problem and effectiveness of query acquisition system and its benefits over an organization.

3. OBJECTIVES AND TECHNOLOGY UTILIZED

- To develop a new system for query acquisition for employees in digital campus and escalating it to the respective teams.
- To Centralize query management: The system should allow for the centralization of all digital campus queries in one location to make it easier for administrators to manage and respond to them.
- To Improve query resolution times: The system should allow for the efficient routing of queries to the appropriate department or individual, reducing response times and improving the overall query resolution process.
- To Enhance communication: The system should be able to provide regular updates on the status of queries, keeping employees informed throughout the query resolution process.
- To Provide data insights: The system should allow for the analysis and collection of data related to campus queries, enabling administrators to identify common issues and trends and make informed decisions to improve campus services.

3.1 TECHNOLOGY UTILIZED

AMAZON WEB SERVICES(AWS)

Amazon Web Services (AWS) is a cloud computing platform provided by Amazon. It provides a wide range of cloud-based IT services and products to businesses, government organizations, and individuals.

AWS SERVICES FOLLOWED:

AWS LAMBDA

AWS Lambda is a serverless computing service from Amazon Web Services (AWS) that allows developers to run code in response to events without provisioning or managing servers. In other words, AWS Lambda allows

developers to write code in various programming languages such as Node. Updates DynamoDB tables. One of the main benefits of using AWS Lambda is that it provides automatic scaling of computing resources, so developers don't have to worry about managing the underlying infrastructure. As AWS lambda helps majorly in this project to write code in python and automate it whenever it is scheduled.

AWS API GATEWAY

AWS API Gateway is a fully managed service from Amazon Web Services (AWS) that enables helps us in this project to create, publish, and manage APIs (application programming interfaces) for their applications.

API Gateway provides a range of features and capabilities, including:

API Creation: API Gateway enables developers to create RESTful APIs supporting both HTTP and HTTPS endpoints, with configurable methods, parameters, request validation and response mapping.

API Deployment: API Gateway enables developers to deploy APIs in multiple stages (e.g., development, test, production), with version control and rollback, and integration with AWS services such as AWS Lambda, Amazon S3 and Amazon DynamoDB.

API Management: API Gateway enables developers to manage APIs through monitoring, statistics, logging, tracing, security, and access control through features such as API keys, IAM roles, and OAuth 2.0.

AWS RELATIONAL DATABASE SERVICE

Relational Database Service (RDS) is a managed database service that allows to deploy and manage relational databases in the cloud for this project. It supports various database engines like Amazon Aurora, MySQL, PostgreSQL, Oracle, and SQL Server.

With AWS RDS, users can easily scale up or down their database instances, automate backup and recovery, and monitor performance metrics. AWS RDS provides features such as automatic software patching, replication, and Multi-AZ deployments to improve database availability and reliability.

AWS CLOUDWATCH

Amazon Web Services (AWS) CloudWatch is a monitoring and observation service that provides real-time information about resources and applications running in the AWS cloud. It helps users monitor and troubleshoot their infrastructure and applications by collecting and analyzing logs, statistics, and events. CloudWatch allows to monitor their AWS resources such as EC2 instances, RDS instances, and Lambda functions whenever necessary in this project. Users can also use the CloudWatch API or SDK to collect metrics and logs from their custom apps and services.

TERRAFORM

Terraform is an open-source infrastructure-as-code (IAC) tool that allows users to define, manage, and provision infrastructure resources across multiple cloud providers, including AWS. Terraform uses a declarative configuration language to define infrastructure resources and dependencies, and to apply those configurations to create and manage resources.

Here In this project, terraform allows users to define infrastructure resources such as Amazon Elastic Compute Cloud (EC2) instances, Amazon Simple Storage Service (S3) buckets, and Amazon Virtual Private Cloud (VPC) networks. Terraform also allows users to manage resource dependencies, allowing them to ensure that all resources required by an application are properly provisioned and configured.

FLASK

Flask is a lightweight Python web framework that allows developers to build web applications quickly and easily. Flask is a micro-framework, which means it only provides the basic tools and functionality needed to build web applications, while allowing developers to customize and extend its functionality as needed. Flask is known for its simplicity and ease of use. It provides a simple yet powerful routing system that maps URLs to display functions, allowing developers to create clean and readable code. As it also supports Jinja2 templates, allowing developers to create dynamic web pages that could be more suitable.

Below chart (Figure 3.1 Architecture Diagram) gives a glimpse on the end- to-end process carried out during the entire project work.

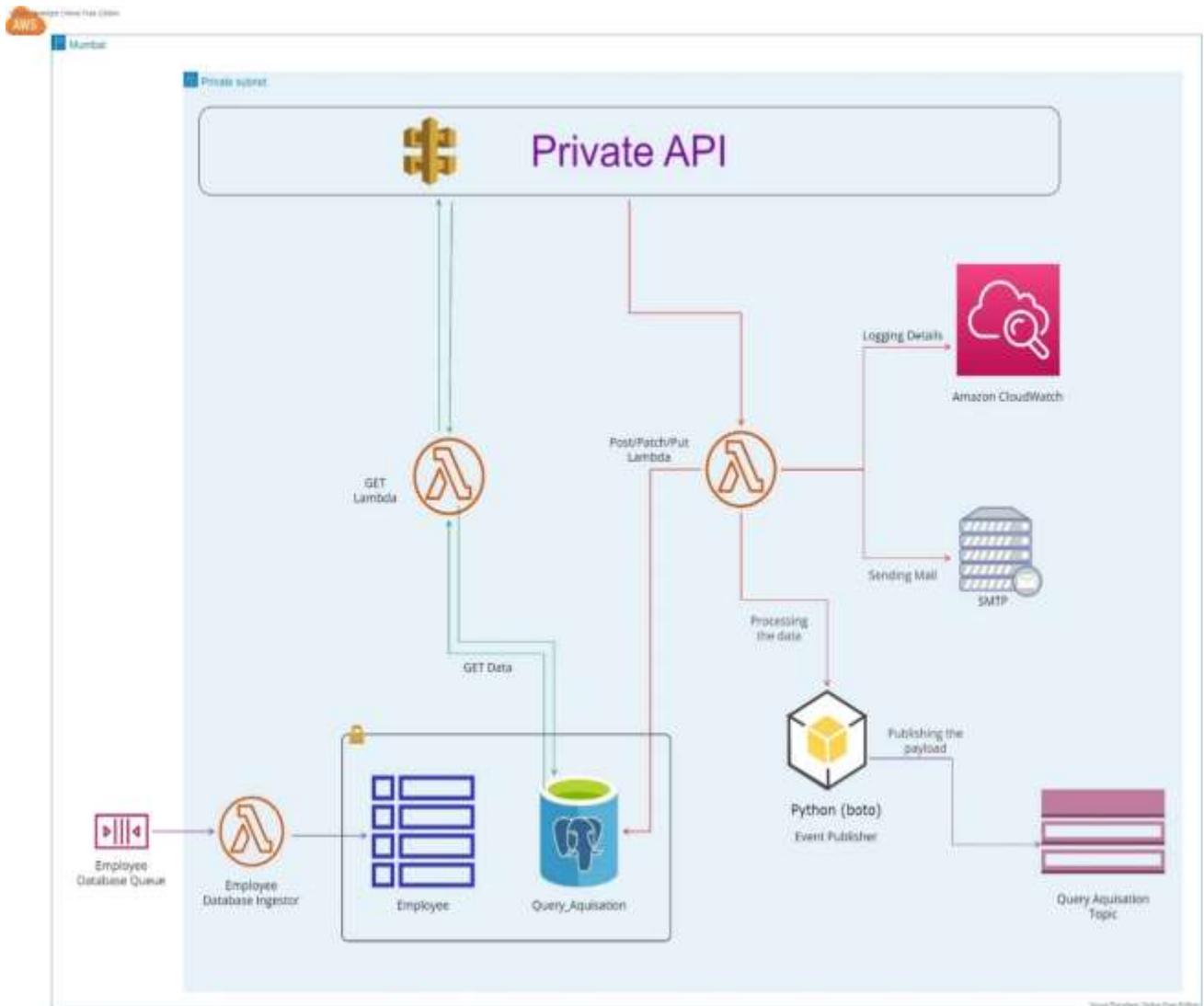


Figure 3.1 Architecture Diagram

3.2 METHODOLOGY

The architecture flow for the digital query acquisition system is shown in Fig 3.1. Initially, the services and AWS account is implemented inside the Mumbai region because most of the account is implements near Singapore to resolve the latency issue with Singapore region here using the Mumbai region.so when that region people using this there will be a less latency among them.Then the services and lambda are added under the private subnet for security purpose. So, the lambda and API cannot be used publicly and can't be used by other resources without specific permission.

The API which is used here in Private API and these API cannot be used by other resources without the access token. In the API there is four resources and under each resource there is 1 method and totally there will be four methods which is using. The methods that is using are 1 POST, 2 GET, and 1 PUT. For each method there is a separate lambda where the python logic will be written.



Figure 4.2 AWS API

- The GET method lambda is used to GET the queries which is created by the employees. The lambda is created because to view the specific and list of queries to log their efforts.
- The POST method lambda is used to POST or insert the new queries by the employee. It is the primary lambda where all the issues are created, and further processing will happen. The details will be store in the database (RDS).
- The PUT method lambda is used to update the request by the specific team when the query or request is opened by the employee. The important process of the lambda is to close the request and log the time spent by the team to resolve the query.
- The process which is happening through the above lambda will be log in the cloud watch. Which is used to verify the issues or errors if raises in the lambda flow.
- The SMTP is used for sending mail which will be connected to the POST and PUT lambda.
- Finally, the details will be published to the simple notification system (SNS) which will be used by other micro service if needed.

3.3 PROCESS FLOW

Flow of the Query Acquisition System:

- Initially when the employee is creating a query via the form the details will be stored in the relational database management system (RDS).
- The details are stored in DB using the postgre sql.
- Once the employee details and queries stored in the Database then the request will open in other table. When the request is opened then the employee will receive a Mail as request is opened and request is update which will be sent when specific team receive a request.
- The specific team where employee raise a query will also receive a mail as request was opened along with the employee details in that mail.
- Once they receive it will be redirect to specific employee and the query will be solved through online mail or with the Microsoft team.
- When the query gets solved and there are no issues by employee then the request will be closed by the team member who solve the issue in the specific form.
- Finally, the closed mail will be sent to the both of them by the query acquisition team as request gets closed.

The backend API and AWS will be connected to the front-end web application through the FLASK web framework where the front-end design is created by using HTML, CSS and JavaScript. From the flask the main tool which is used is jinja2 template engine which is used to send the dynamic data to the HTML and CSS.

4. CONCLUSIONS

As In Digital Campus there is no proper and appropriate ways to raise queries and complaints. As The Digital Campus Query Acquisition System for digital campus projects is a system that enables employees, co-workers, and staffs to access information and resources necessary for their daily activities. It is designed to be a one-stop-shop for all queries related to campus activities, including access requirements, Administer access, and general queries. The system is user-friendly and accessible, making it easy for users to navigate and find the information they need quickly.

In conclusion, the digital campus query acquisition system is an essential tool for managing queries and improving communication within an organization. The system's user-friendly interface, responsiveness, and accessibility make it an invaluable asset for organizations of all sizes. The system enhances collaboration among different departments and units, improves customer service, and provides a comprehensive platform for managing different aspects of the organization's activities. Overall, the query acquisition system is an essential tool for organizations looking to improve efficiency, productivity, and effectiveness.

5. REFERENCES

- [1] Awa, H.O., Ojiabo, O.U. and Orokor, L.E., 2017. Integrated technology-organization-environment (TOE) taxonomies for technology adoption. *Journal of Enterprise Information Management*.
- [2] Bayrak, T., 2013. A decision framework for SME Information Technology (IT) managers: Factors for evaluating whether to outsource internal applications to Application Service Providers. *Technology in Society*, 35(1), pp.14-21.
- [3] Dutta, P. and Dutta, P., 2019. Comparative Study of Cloud Services Offered by Amazon, Microsoft & Google. *International Journal of Trend in Scientific Research and Development (ijtsrd)*, 3, pp.981-985.
- [4] Gandhi, A. and Chan, J., 2015. Analyzing the network for aws distributed cloud computing. *ACM SIGMETRICS Performance Evaluation Review*, 43(3), pp.12-15.
- [5] Kirchgaessner, S., 2013. Cloud storage carries potent security risk. *Financial Times*. Viitattu, 31, p.2014.
- [6] Layo, I. 2013. Cloud Computing Advantages for SMEs. Available online: <http://cloudtimes.org/2013/09/18/cloud-computing-advantages-for-smes/> (accessed on 6 December 2020).
- [7] Mosca, P., Zhang, Y., Xiao, Z. and Wang, Y., 2014. Cloud security: Services, risks, and a case study on amazon cloud services. *Int'l J. of Communications, Network and System Sciences*, 7(12), p.529.