GETMEDS: Online Pharmacy Management Application
Using AWS Cloud Technologies

COEN 241 - CLOUD COMPUTING - WINTER 2023

Sumit Agrawal	Arpita Verma	Steve DCosta	Rohnie Baskar
SCU ID: W1631657	SCU ID: W1632653	SCU ID: W1606759	SCU ID: W1629049
Computer Science	Computer Science	Computer Science	Computer Science
Santa Clara University	Santa Clara University	Santa Clara University	Santa Clara University

***Abstract*— This paper presents a comprehensive study on the development and deployment of a pharmacy management web application over the cloud using Amazon Web Services (AWS). The web application was built using the Django web framework and Python programming language. This report highlights the key features of the pharmacy management web app, including prescription processing, patient record management, inventory management, and report generation. The system is designed to be highly scalable, secure, and available 24/7. We conclude with a detailed analysis of the benefits of using cloud computing for pharmacy management systems, including improved flexibility, cost-effectiveness, and scalability. We also discuss the potential impact of our work on the healthcare industry, particularly on improving patient outcomes and reducing operational costs for pharmacies.**

Keywords*— *Pharmacy management, AWS Cloud services, web-based application, Django, Scalable application

I. INTRODUCTION

We have developed a pharmacy management system - a software application that helps streamline the daily operations of a pharmacy. Django is a popular web framework that was used to build a robust and scalable web application quickly and efficiently. By combining Django with cloud computing technology, pharmacists can access their pharmacy management system from anywhere in the world, providing greater flexibility and convenience.

This pharmacy management system built using Django can help pharmacists manage inventory, track prescriptions, and medication orders, and generate reports to monitor performance and compliance. By leveraging cloud computing, this system can be easily scaled up or down to meet changing business needs, while also providing a high level of security to protect sensitive patient data.

Deploying a web-based pharmacy management system over the cloud can offer several benefits, such as increased accessibility, scalability, and cost-effectiveness. With the ability to access the system from anywhere, pharmacists can quickly respond to patient needs and manage their pharmacy operations more efficiently. Additionally, cloud-based systems are often more cost-effective than traditional on-premise solutions, as they require less hardware and IT support.

Thus, deploying over the cloud can help pharmacies improve their operational efficiency, reduce errors, and enhance patient care, making it a valuable tool for any pharmacy looking to streamline its operations and stay competitive.

This report is structured as follows. In the next section, we discuss the motivation behind this project, especially about a web based application over cloud. In section III we discuss the related and past work about this topic. In section IV we describe in detail the application, its framework and the system design plan. In section V we describe the step by step approach of our System Development Life Cycle, the tech stack and the architecture. The next part will talk about the advantages of having the application over cloud and why we decided to deploy our system over the AWS platform based cloud. The Use case is described in section VII for understanding the frontend and backend structure of the software. Section VIII will provide details of teh cloud technologies we used specifically for our product. In the next part, we talk about the user interface of our app-GetMeds. Section X will provide the outcome of our project. Section XI will give insights into the future work that can be implemented in our application and then we conclude our report in the last section.

II. MOTIVATION

Federal legislation mandates the employment of permanent systems (digital or not), which continually record the number of prescriptions filled and issued as Schedule II controlled medicines. In this manner, the drug is automatically taken out of stock, and you always have up-to-date stock information.*

Several patients needed medications urgently during the Covid (Pandemic), which would have been impossible to manage with standard storage devices and a single system. In this situation, a cloud-based solution would make it scalable and be helpful during the surge conditions.

Even when medication counts are performed on a regular basis, sometimes the numbers are inaccurate, or the system is not updated in a timely manner. In this scenario, a pharmacy management system can be utilized to maintain a thorough track of our inventory that is easily filterable by the necessary storage conditions and expiration date, allowing us to avoid potentially harmful mistakes.

Since the chance of human error is reduced, this reduces paperwork and helps ensure that the order is never lost or misunderstood. The dosage is always accurate, the patients' information is maintained in a database, and the doctor's data is available and can be contacted by the pharmacists in case of any requirements.

These various factors contributed to our recognition of the significance of comprehending the underlying system of a pharmacy application and its deployment over the cloud. Consequently, we resolved to acquire knowledge and deploy our web application over the cloud, resulting in a comprehensive understanding of fundamental Django programming concepts for a web application and utilization of AWS cloud technologies for cloud deployment.

III. BACKGROUND AND RELATED WORK

Though the pharmacy management system has been in place for a while, our application aims to incorporate both the online ordering of medicines along with delivery and online doctor consultation.

Work from [1] has created an application to effectively manage the stocking of drugs thereby improving efficiency as well as a reduction in manual error which is very crucial in the medical field since a mistakenly administered expired drug can have significant repercussions. We have implemented this idea to manage the stocking as well as the expiration of drugs in our application. Work from [2] deals with online prescription management systems using IOT. It results in a better diagnosis of underlying illness since the scope for error is reduced and the maintenance is easier. The liberty pharmacy management system[5] has enabled us to design a single application for all the modules. The software helps in generating reports, handling refills, managing insurance, managing pharmacies as well as documentation.

The related work from [1], [2], and [5] has inspired us to come up with the pharmacy management system and then as a stretch goal to implement the ordering and online consultation as well.

IV. DESCRIPTION

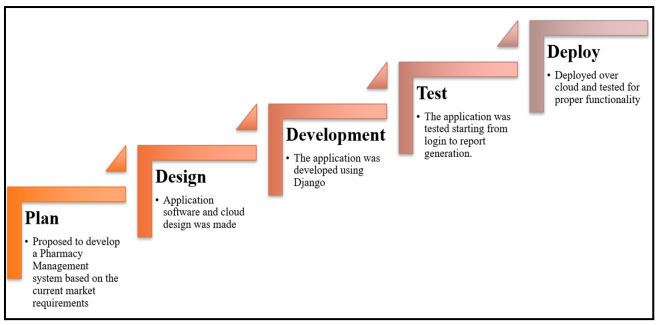
Our team has developed a pharmacy management system using Django and Python, which will include both frontend and backend components. The application will have a login page for authorized administrators to access the system, where they can create, view, edit, or delete data, as well as generate reports.
Django and Python are popular and reliable technologies for building web applications, allowing for rapid development, flexibility, and scalability. Our application will utilize Django's built-in admin panel for easy data management, and will also incorporate custom views for more specialized functionality.

To ensure high availability and scalability, we plan to deploy the application over the cloud using AWS. AWS offers a range of cloud computing services that are well-suited for web application deployment, including Amazon EC2 for hosting virtual servers.

V. APPROACH

SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle (SDLC) is a standardized process for developing and deploying software applications. This SDLC follows the traditional approach, which includes the following stages for our application for pharmacy.



Plan: In this stage, the project goals and requirements were identified, and a plan was proposed to develop a Pharmacy Management system based on the current market requirements. The plan was created with a clear understanding of the scope, objectives, and limitations of the project.

Design: In this stage, the application software and cloud design were made. The system architecture and components were defined, and the user interface and user experience design were created. The design was created to ensure that the application meets the requirements of the project plan.

Development: In this stage, the application was developed using Django, a popular web framework that allows for rapid and efficient development. The development process included coding, testing, and debugging. The development team followed a set of coding standards and best practices to ensure that the code is maintainable, scalable, and robust.

Test: In this stage, the application was tested starting from login to report generation. The testing process was carried out to ensure that the application meets the functional and non-functional requirements of the project plan. The testing team used a range of testing techniques and tools to identify and fix any defects or issues.

Deploy: In this stage, the application was deployed over the cloud and tested for proper functionality. The deployment process involved configuring the cloud infrastructure, setting up the database, and deploying the application. The deployment was done using AWS services, which allowed for scalability and high availability.

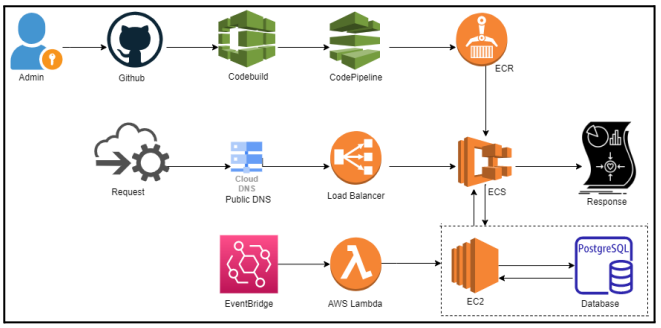
Overall, this SDLC ensured that the Pharmacy Management system was developed efficiently and effectively, meeting the requirements of the project plan. The SDLC provided a structured approach to the development process, which helped to ensure that the application was delivered on time.

TECH STACK

- Frontend: HTML, CSS, Django Template Language
- Backend: Django, Python
- AWS ECS
- Docker image: AWS ECR
- Postgres Database configured on AWS EC2
- Backup:
    - AWS Lambda
    - AWS Event Bridge
- CI/CD: Code build, Code pipeline.

ARCHITECTURE

The web application we have developed uses a combination of HTML, CSS, and the Django Template Language to provide an intuitive and user-friendly frontend interface. The backend of the application is built using the Django web framework and Python programming language, which allows for efficient and scalable development.



To deploy our web application over the cloud, we have used Amazon Web Services (AWS) Elastic Container Service (ECS) to manage the containers that run our application. Our application's Docker image is stored on AWS Elastic Container Registry (ECR), which provides secure and scalable storage for our container images.

For our database, we have configured a Postgres database on AWS EC2, which allows us to store and manage our data securely in the cloud. To ensure data resilience and continuity, we have implemented a backup solution using AWS Lambda and AWS Event Bridge. This backup solution automatically triggers a backup of our data at regular intervals and sends notifications in case of any failures.

To streamline our development and deployment process, we have implemented a Continuous Integration/Continuous Deployment (CI/CD) pipeline using AWS Code Build and AWS CodePipeline. Our CI/CD pipeline allows us to automate the build, test,

and deployment process of our application, ensuring that we can quickly and easily deploy new features and updates to our users.

Overall, our web application leverages the power of modern technologies, such as Django, Python, and AWS, to provide a robust, scalable, and secure solution for pharmacy management.

By utilizing AWS services such as ECS, ECR, Lambda, and Event Bridge, we can ensure that our application is highly available, resilient, and secure, providing a seamless user experience for pharmacy professionals.

VI. WHY CLOUD?

In a cloud-based pharmacy management system, backups are performed automatically and scheduled to run in the background without requiring manual intervention. This is a more practical solution compared to manual procedures, which can be time-consuming and prone to errors. With cloud-based backups, you can be assured that your critical data is backed up on a regular basis, minimizing the risk of data loss in case of unforeseen events such as hardware failures or cyber-attacks.

Another advantage of a cloud-based pharmacy management system is that the cloud providers take care of server maintenance. This means that you don't have to worry about hardware failures or software updates, as they are managed by cloud providers. Cloud providers ensure that their servers are always up and running and that any issues are quickly resolved to minimize downtime. This allows you to focus on managing your pharmacy and serving your customers, without worrying about the technical aspects of server maintenance.

In the context of pharmacy management, the security and safety of patient and pharmacy data is of paramount importance. Traditional storage devices like external hard drives or USB drives are not always the most reliable or secure way to store sensitive information. These devices can be lost, stolen, or damaged, leading to the potential loss of critical data.

In contrast, cloud storage offers a higher degree of protection for data. The use of cloud-based storage for a pharmacy management app provides additional layers of security that cannot be offered by traditional storage devices. The data is stored offsite in the cloud, adding an extra layer of protection against environmental threats such as theft, floods, and fires in addition to technical issues like hacking.

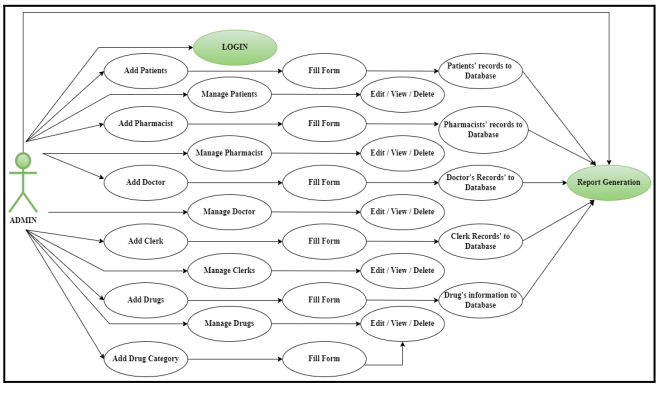
Data is a critical asset that must be protected from loss or corruption here. Cloud-based backups offer several levels of redundancy, which means that multiple copies of the data are stored in different locations. This approach provides added protection against accidental deletion or loss, ensuring that the data can be recovered in case of a disaster or outage.

Furthermore, cloud-based backups are much more convenient and efficient than manual backup procedures. With manual backups, finding and restoring the data can be a time-consuming process. However, with cloud-based backups, the data can be easily located and restored with a quick call to the customer support team.

In addition to providing enhanced security and ease of use, cloud-based backups also offer cost-effective benefits. With manual backup procedures, the cost of hardware, storage media, and transportation must be factored in. However, with cloud-based backups, there is no need to invest in any additional hardware or equipment, which can save significant costs over time.

By leveraging cloud technology, servers can be optimally utilized without any server being left idle, resulting in cost savings and increased efficiency. This is because cloud providers have the capability to scale server resources up or down based on demand, ensuring that the pharmacy management system runs smoothly at all times, without any bottlenecks or delays. This dynamic allocation of resources enables businesses to pay for only what they need, eliminating the need for expensive hardware and manual maintenance. Furthermore, cloud-based systems offer greater flexibility in terms of storage, as businesses can quickly add or remove storage as per their requirements. Overall, the use of cloud technology for pharmacy management systems results in better resource utilization, increased efficiency, and reduced costs.

VII. USE CASE



A Use Case diagram in Unified Modeling Language (UML) is a graphical representation of the different ways users can interact with a software application to achieve specific goals or tasks. In the context of our pharmacy management software application that contains a login for admin and pharmacists, adding, viewing, and editing data, and generating reports, the Use Case diagram helps us to visualize the different functionalities of the application and the interactions between the users and the system.

The Use Case diagram for this pharmacy management software application would include the following use cases:
The actors for all these use cases would be the admin, doctors, clerks, and pharmacists.

- Login: This use case represents the login functionality of the application, which allows authorized users to access the system.
- Add product: This represents the functionality to add new products to the inventory.
- Edit product: This represents the functionality to modify the information related to a specific product.
- View inventory: This represents the functionality to view the inventory of the pharmacy.
- View patient information: This represents the functionality to view patient information, including prescription history, allergies, and medical conditions.
- Generate report: This represents the functionality to generate reports based on the data in the system.

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import *

class PatientsAdmin(admin.ModelAdmin):
    list_display = ('admin','gender')
    search_fields = ["admin__username"]
class UserModel(UserAdmin):
    pass
admin.site.register(CustomUser, UserModel)
admin.site.register(Patients,PatientsAdmin)
admin.site.register(Pharmacist)
admin.site.register(AdminHOD)
admin.site.register(Stock)
admin.site.register(Category)
admin.site.register(Doctor)
admin.site.register(PharmacyClerk)
admin.site.register(Prescription)
admin.site.register(Dispense)
admin.site.register(PatientFeedback)
```

Code snippet from our project

The Use Case diagram would also illustrate the relationships between the different use cases and actors. For example, the Login use case would be connected to all other use cases, indicating that all use cases are dependent on the user being authenticated and authorized to use the system. The Add product and Edit product use cases would be connected to the View inventory use case, indicating that these use cases are dependent on the availability of the inventory. The Generate report use case would be connected to all other use cases, indicating that the reports generated would be based on the data entered and managed by the other use cases.

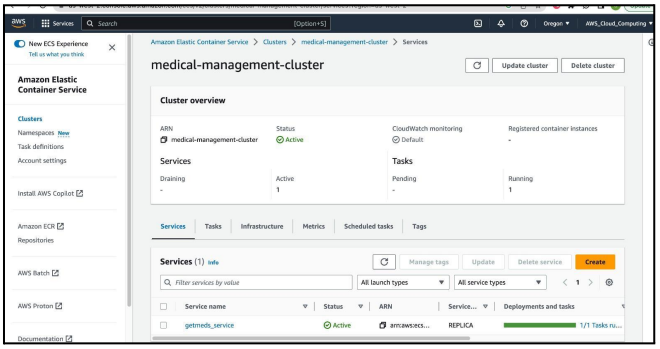
In conclusion, the Use Case diagram for a pharmacy management software application containing login for admin doctors, clerks and pharmacists, adding, viewing, and editing data, and generating reports would help to illustrate the different functionalities of the application and the interactions between the users and the system.

VIII. CLOUD TECHNOLOGIES USED

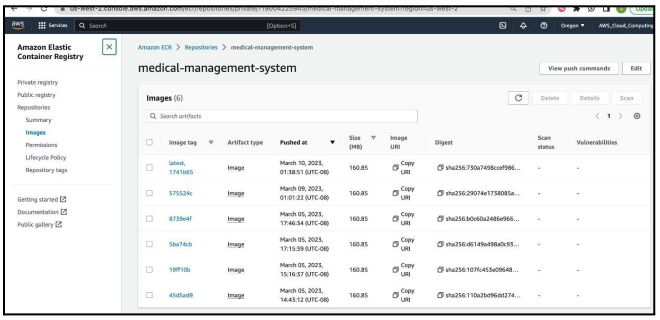
Here we provide the details of the AWS cloud technologies that we used with the screenshots of the deployment.

Amazon ECS - Amazon ECS (Elastic Container Service) is a fully managed container orchestration service provided by Amazon Web Services (AWS). It allows users to easily run, manage and scale Docker containers in a cluster of EC2 instances or with AWS Fargate, which is a serverless compute engine for containers.

With Amazon ECS, we can deploy and manage containerized applications at scale. It provides features such as auto-scaling, load balancing, and automatic container recovery to ensure high availability and scalability of containerized applications.

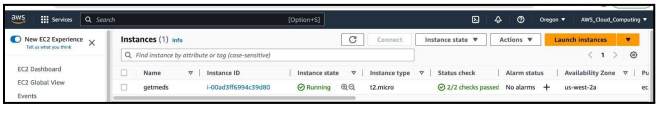


Docker Image: Amazon Elastic Container Registry (Amazon ECR) - Amazon Elastic Container Registry (Amazon ECR) is a fully-managed Docker container registry provided by Amazon Web Services (AWS). It allows users to store, manage, and deploy Docker images in a secure and scalable way.

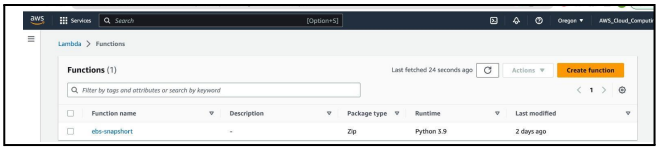


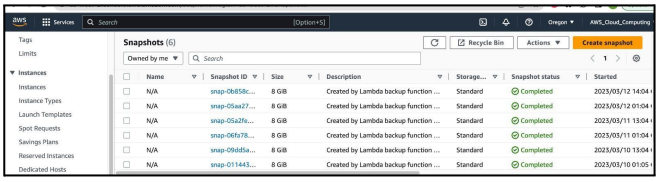
Amazon EC2 - Amazon EC2 (Elastic Compute Cloud) is a web service provided by Amazon Web Services (AWS) that allows users to rent virtual machines on the cloud. With Amazon EC2, users can easily launch virtual servers, also known as instances, and run any software application of their choice.

Amazon EC2 provides a highly scalable and flexible compute capacity, allowing users to easily scale up or down based on demand.



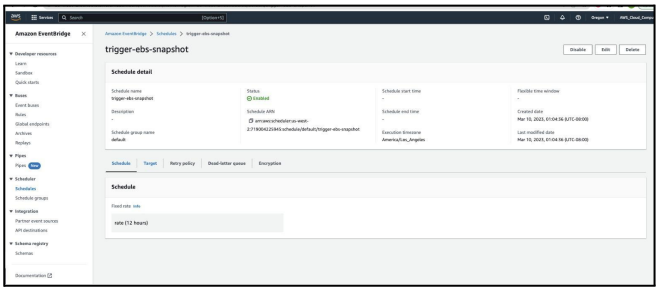
Amazon Lambda - Amazon Lambda is a serverless compute service provided by Amazon Web Services (AWS). It allows users to run code without the need to provision, manage, or scale servers. With Lambda, users can simply upload their code and specify the amount of memory required to execute it, and Lambda takes care of the rest.





Amazon Eventridge - Amazon EventBridge is a serverless event bus provided by Amazon Web Services (AWS) that makes it easy to build event-driven architectures. With EventBridge, users can create event-driven workflows by integrating various AWS services and SaaS applications.

EventBridge allows users to collect events from various sources, such as AWS services, partner SaaS applications, and custom applications. It provides a central place to manage these events, making it easy to route them to different targets based on rules and policies.

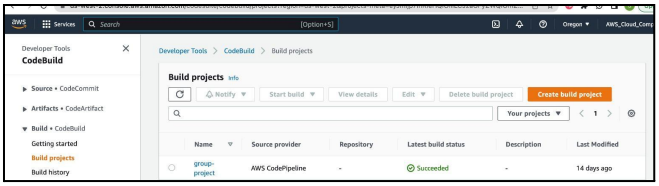


AWS CodeBuild - AWS CodeBuild is a fully managed continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS). It allows users to build, test, and deploy code automatically based on the predefined configurations and workflows.

With AWS CodeBuild, users can build code in various programming languages, such as Java, Python, Ruby, and more. It integrates with other AWS services, such as AWS CodeCommit, AWS CodePipeline, and AWS CodeDeploy, making it easy to build complete CI/CD pipelines.

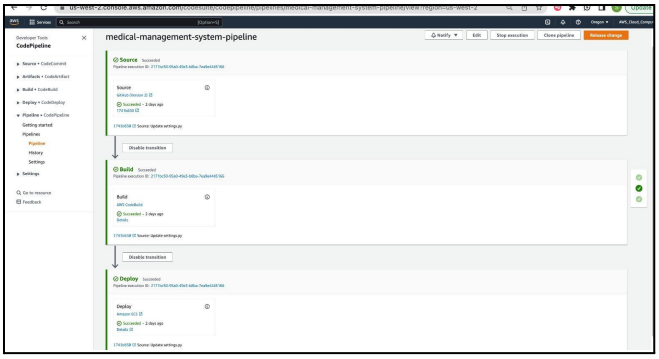
AWS CodeBuild provides a wide range of features to manage the build process, such as custom build

environments, caching, and parallel builds. It also provides integration with other third-party tools such as Github.



AWS CodePipeline - AWS CodePipeline is a fully managed continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS). It allows users to build, test, and deploy code automatically based on predefined configurations and workflows.

With AWS CodePipeline, users can create automated pipelines for building and deploying code in various environments, such as development, testing, and production.



IX. APPLICATION USER INTERFACE

The user interface (UI) of our pharmacy management application includes various features and functionalities to help users manage pharmacy-related tasks easily and efficiently. Here are some key UI elements and their functions:

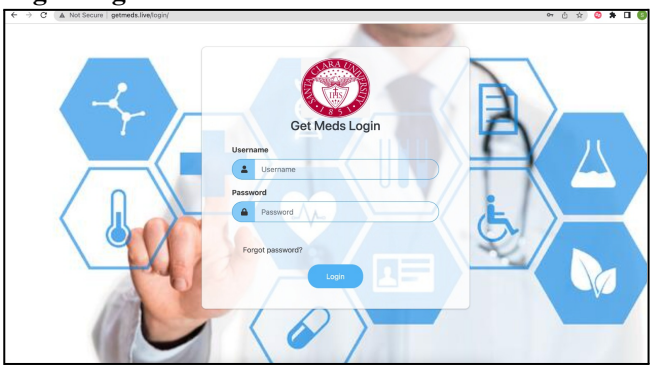
Login and authentication: Our application has a login page that requires users (admin, doctors, pharmacists, clerks) to enter their username and password to access the system. This helps ensure that only authorized personnel can access sensitive information.

Dashboard: The dashboard provides an overview of the key information and functions available in the application, such as inventory levels, prescription orders, and patient information.

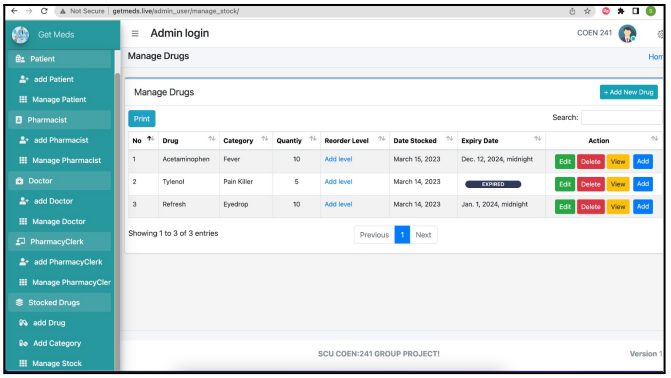
Inventory management: This application allows users to manage inventory levels, track expiration dates, and reorder items when necessary. This can include features such as adding new products, editing existing records, and generating reports on inventory levels.

Reporting and analytics: Our system provides users with tools to generate reports and analyze data related to pharmacy operations, such as employee data, patients' demographics, and inventory levels.

Login Page of Getmeds.live



Application User Interface:



X. OUTCOME

One of the key outcomes of using GETMEDS is improved efficiency in pharmacy management. The app streamlines inventory management, and tracking, allowing pharmacies to easily manage their stock and avoid stockouts. Additionally, the app automates many of the processes involved in prescription management, including electronic prescribing, and refill requests, reducing the workload on pharmacy staff and improving the accuracy and speed of prescription processing.

Moreover, the cloud-based nature of GETMEDS allows for real-time data access and analysis, providing pharmacies with valuable insights into their operations and customer behavior. This information

can be used to make data-driven decisions and improve the overall performance of the pharmacy.

After the successful deployment of our pharmacy management application on the cloud, we conducted various load-testing scenarios to evaluate its performance under different traffic and usage conditions. The results of these tests were analyzed and collated, and we are pleased to report that the application demonstrated exceptional performance across the board.

In particular, we recorded the application's response time and resource utilization across low and medium traffic and usage conditions. Under low traffic conditions, the application showed an impressive response time of just 50 milliseconds, while in medium traffic conditions, this response time increased to 80 milliseconds.

Furthermore, in low usage conditions, the application consumed only 50 MB of resources, while in medium usage conditions, it consumed 100 MB. These results showcase the application's ability to handle varying levels of traffic and usage, while simultaneously minimizing the consumption of resources.

We recognize that ensuring the optimal performance of a pharmacy management application is critical to the success of any pharmacy business. Therefore, our application on the cloud was tested and proven to deliver exceptional performance in varying traffic and usage conditions.

Thus, our pharmacy management application on the cloud is equipped to handle varying levels of traffic and usage while delivering fast response times and minimal resource utilization.

Deploying this pharmacy management app on the cloud using Amazon Web Services (AWS) technologies provided us with a number of valuable outcomes. One of the main outcomes is the ability to learn the basics of AWS technologies and how to use them effectively for application deployment.

AWS offers a range of services that can be leveraged for deploying an application on the cloud, including Amazon Elastic Compute Cloud (EC2), Amazon Lambda, Amazon Codebuild and Codepipeline, and AWS Eventrbridge. By using these services, we get many benefits of cloud computing, such as scalability, flexibility, and cost-effectiveness.

The process of deploying a pharmacy management app on the cloud involves setting up an AWS account, selecting the appropriate services, and configuring the app to work with those services. This process has provided valuable learning opportunities for us, including gaining experience with cloud technologies and understanding how to use them to deploy and manage an application.

XI. ANALYSIS & FUTURE SCOPE

Our pharmacy management application can be enhanced with the addition of an online ordering and delivery system for medicine. This new feature will allow customers to conveniently place orders for their required medication through our application and receive the timely delivery of the same. Furthermore, this system will enable secure payment options to streamline the entire ordering and payment process. Our aim is to provide our customers with a seamless and hassle-free experience while ensuring the safety and efficacy of their medication.

Our pharmacy management app will incorporate an advanced information-gathering system that will enable us to conduct additional analysis and provide enhanced services to our patients. By capturing and recording a patient's medical history and symptoms, our analytics models can accurately identify potential health issues and recommend the most effective and safe medications for treatment. This feature will provide our patients with personalized and accurate medication recommendations, thereby improving their overall health outcomes. Our commitment to leveraging technology to enhance patient care is at the heart of this feature, and we believe it will further strengthen our position as a trusted healthcare provider.

Our pharmacy management app can be equipped with a Patient Care Program, which will utilize cutting-edge medication adherence tools such as notifications and calendars to ensure patients are adhering to their prescribed medication regimen. This program fosters collaboration between physicians and pharmacists, who work together to create a customized medication therapy plan that is tailored to the unique needs of each patient. By closely monitoring medication adherence and tracking the effectiveness of the prescribed therapy, we can improve patient outcomes and reduce the risk of adverse events.

XII. CONCLUSION

This system can bring numerous benefits to the pharmacy industry. The system can help pharmacy

owners to streamline their operations, manage inventory efficiently, and improve customer service.

The application can also be equipped to process real time data with the help of cloud technologies, thereby enabling the business to take data driven decisions as well aid in predicting, diagnosing and curing proactively

Overall, the Pharmacy Management System project has the potential to revolutionize the way pharmacies operate, making them more efficient, effective, and profitable. With the right implementation strategy and ongoing support, this system can be a game-changer for the pharmacy industry, helping to improve patient outcomes and customer satisfaction while driving business growth.

We have developed an end-to-end application 'GETMEDS' which uses cloud technologies for all of its infrastructure. We implemented multiple modules like patient, doctor, pharmacist, clerk and drug management for our application. On the cloud side we realized the various advantages of cloud technology like scalability, reliability, convenience and optimized use of resources. As a future goal we intend to add the online ordering as well as online consultation to our application as well making it a one stop application for all medical needs.

REFERENCES

[1] U. Rivett et al., "A pharmacy stock control management system to effectively monitor and manage patients on ART," 2006 The 4th Institution of Engineering and Technology Seminar on Appropriate Healthcare Technologies for Developing Countries, London, UK, 2006, pp. 27-36, doi: 10.1049/ic.2006.0657.

[2] A. Mehmood, F. Mehmood and W. -C. Song, "Cloud based E-Prescription management system for healthcare services using IoT devices," 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2019, pp. 1380-1386, doi: 10.1109/ICTC46691.2019.8939916.

[3] R. T. Hameed, O. A. Mohamad, O. T. Hamid and N. Tapus, "Design of e-Healthcare management system based on cloud and service oriented architecture," 2015 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 2015, pp. 1-4, doi: 10.1109/EHB.2015.7391393.

[4] https://aws.amazon.com

[5] https://libertysoftware.com/pharmacy-software/

[6] https://nix-united.com/success_stories/cloud-based-pharmacy-management-system-for-increasing-efficiency-and-improving-patient-care/

Github Link to our project - https://github.com/sumit8698/COEN241-Cloud-Computing/tree/main/Project