

# DEVELOPING A PLASMA DONOR APPLICATION

## ABSTRACT:

A plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish antibodies that fights the infection. In this project plasma donor application is being developed by using AWS services. The services used are AWS Lambda, API gateway, DynamoDB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve serverless computing. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates were high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

## LITERATURE SURVEY:

In this paper, the author has carried out analysis based on the opportunities presented by serverless computing. They emphasise that serverless services are more affordable approach for many network services and it is more user friendly as serverless approach will relieve the customers from the intricacies of deployment. These services will help to improve the new business opportunities.

[2] Author conducted a survey of existing serverless platform in this paper from source projects, industry, academia, use cases, and key characteristics and has described the challenges and the open problems associated with it. Authors work presented a hands-on experience of serverless technologies using different services from different cloud providers such as Amazon, Google, IBM, Microsoft Azure.

[3]In this paper three demonstrators for IBM Bluemix OpenWhisk was presented. They exhibit even-based programming triggered by weather forecast data, speech utterances and Apple WatchOS2 application data. And also demonstrated a chatbot using IBM Bluemix OpenWhisk that calls on the IBM Watson services which include dates, weather, alarm services, news and music tutor.

[4]In this paper serverlessOS was designed. It comprises of components such as 1. desegregation model that leverages desegregation for abstraction but it will enable resources to move fluidly between servers for the performance. 2. The second key component is cloud orchestration layer which helps to manage fine-grained resource placement and allocation throughout the application lifetime with the help of global and local decision making

3. And the third component is an isolation capability which enforces data and resource isolation.

[5]In this paper an efficient resource management system for serverless computing framework was proposed which aims to enhance resource with a focus on memory allocation among the containers and the design which was added on top of an open-source serverless platform, openLambda and it is based on allocation workloads and serverless functions memory needs events are triggered.

## Flow of Mechanism:

User will first register to the web application then will login to the application. If a user can successfully login, then he can check the availability of the blood group and can request for a particular blood group. Once the user requests for a blood group then the concerned blood group donor will receive a request sms.

## REFERENCES:

[1] R. C. Gojko Adzic, "Serverless computing: Economic and architectural impact," ESEC/FSE, 2017.

[2] P. C. P. C. a. V. I. M. Yan, "Building a chatbot with serverless computing," IBM watson research center, 2016.

[3] S. E. a. B. J. J. Short, ““Cloud Event Programming Paradigms: Applications and Analysis,”” 9th IEEE International Conference on Cloud Computing (CLOUD), pp. pp. 400-406, 2017.

[4] Z. Al-Ali, ““Making Serverless Computing More Serverless,”” IEEE 11th International Conference on Cloud Computing (CLOUD), pp. pp. 456-459, 2018., 2018.

[5] A. S. a. S. Jindal, ““EMARS: Efficient Management and Allocation of Resources in Serverless,”” IEEE