International Journal of Research in Engineering, Science and Management
Volume-3, Issue-5, May-2020
www.ijresm.com | ISSN (Online): 2581-5792

Gamification Based Learning

Akarshan Agarwal[1], Shreevaths K. Satish Rao[2*], B. M. Mahendra[3]
[1,2]*Student, Department of Electronics and Communication Engineering, R. V. College of Engineering, Bangalore, India*
[3]*Assistant Professor, Department of Electronics and Communication Engineering, R. V. College of Engineering, Bangalore, India*
*Corresponding author: shreevathsr@gmail.com

***Abstract*: The present paper presents a novel way of learning for the students using a variety of features leveraging the competitive nature of students via the use of weekly and monthly ranking, leaderboard and special mention on the platform on achieving something. The platform is comprised of two modules – Learning and Contest activities. The existing platforms were having the above modules as part of separate applications which created issues mainly with respect to tracking. The proposed application also has a host of features such as creation of meetings in Microsoft Teams and Zoom, conducting other activities such as Daily Scrum Meetings, Assessments etc. The tech stack for the application comprises of ReactJS, Spring Boot, Docker, AWS, Git and testing conducted with Junit5, React Testing Library and Gatling – Backend performance testing, Lighthouse- Frontend Performance testing.**

***Keywords*: Gamification, Tech stack, SCRUM, AWS.**

1. Introduction

The application is an in-house application developed in a software development company to be used to train the incoming interns and new hires which also had the added benefit of making the onboarding process smooth and less time-consuming. The gamification aspect of the application is what makes it unique and ensures that no interest is lost by arousing the competitive nature of the new hires through a system of leaderboard and rankings.

The application is also web-based hence it would involve front end and backend along with cloud hosting. Hence this Gamification Learning Program was a full stack project which uses different technology stacks.

2. Problem Definition

To provide a platform where the learning is made in an extremely engaging manner. The application also provides a gamification aspect where assessments and learnings that have been completed within time are being rewarded extra marks. This helps in engaging a student to learning and will take efforts to make sure that he submits his assessments on time and also the courses which are assigned to user.

3. Objectives

Customer Experience: The application should provide an enriched look and feel that could make an interactive user experience for the users that could bind them to the application and help them continue the learning process on a regular basis.

Scalable Application: The application should be completely scale-able to ensure that it could be developed in any of the regions where the company can ensure that the growing number of users could be added at any point of time and also extra functionality could be added [4].

Availability: The application availability should be high i.e. should be operational, functional and usable for competing or fulfilling a user's and business requirement [6, 7].

Robustness and Resilience: The application should be robust [5] to handle all kind of errors and handle them in the correct possible manner. In case of failure the alerting method should be present and a possible recovery procedure should be in place that could help the application in.

4. Design of Architecture

The application project was based on Spring Framework and followed the Netflix OSS Architecture. Spring Boot helped in the setting up of the given architecture and the various components were the key components. Along with the MVC layered architecture [1] was also followed which helped in the flow of data via controllers, transactions services, repository layer and finally the database. The components listed in the above diagram are explained briefly as followed,

Configuration Server: Spring Cloud Config Server provides an HTTP resource-based API for external configuration (name-value pairs or equivalent YAML content). The server is embeddable in a spring application. Pairs or equivalent YAML content). The server is embeddable in a Spring Boot application, by using the @EnableConfigServer annotation. Config Server files can either be stored in local files, Git files or environment setup hosted on server.

Zuul Server: Zuul Server is a gateway application that handles all the requests and does the dynamic routing of micro service applications. The Zuul Server is also known as Edge Server. For example: /api/user is mapped to the user service and /api/products is mapped to the product service and Zuul Server dynamically routes the requests to the respective backend application. The Zuul Server is also enabled with a load balance

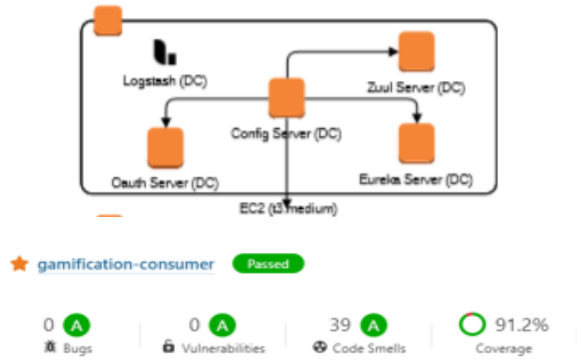
which helps in load balancing of calling and consuming microservices. This load balancing can be of client side balancing as well as server side balancing. Zuul is auto enabled with client side load balancing know as Ribbon Client Load Balancer. Feign Client It is a declarative HTTP client developed by Netflix. Feign aims at simplifying HTTP API clients.

Eureka Server: The Eureka server is nothing but a service discovery pattern implementation, where every micro service is registered and a client micro service looks up the Eureka server to get a dependent micro service to get the job done. The Eureka Server is a Netflix OSS product, and Spring Cloud offers a declarative way to register and invoke services by Java annotation.

5. Result

The application is based on TDD approach. The sonar is implemented as part of deployment service, this leads to coverage of 91.2% with 39 code smells as shown in Fig. which ensures code quality and that new code added each iteration has been thoroughly tested before deployment.

The application is been deployed on AWS which offers high availability across different zones. It is monitored in real-time using ELK stack. The OAuth 2 is implemented as part of security which handle 250 concurrent request at a time. All the requests involved in the application occur over TLS which provide security for all communication involved. The data integrity is ensured by following commit-all or none approach along with data normalization. The delivery service of the application (Kafka) has a scheduler module which runs round the clock and ensures guaranteed delivery of messages in the event of failure. The front-end performance has been tested with lighthouse and is within acceptable range.



Kafka uses a binary TCP-based protocol that is optimized for efficiency and relies on a "message set" abstraction that naturally groups messages together to reduce the overhead of the network roundtrip. The consumer and producer APIs build on top of the Kafka messaging protocol and offer a reference implementation for Kafka consumer and producer clients in Java. The underlying messaging protocol is a binary protocol that developers can use to write their own consumer or producer clients in any programming language.

Assigning the learning or assessment Functionality:

The main base of the application is to provide different type of activities to a user such as a learning course, an assessment or a webinar. The admin gets the access to create these types of activities, provide details to it and also assign them to an individual user, an entire batch else an entire category of user. Along with it the admin can also assign activities from Udemy and contest from Hackerank which will be integrated in the system. The admin can assign some assessment rules to a contest which will help in the leaderboard structure.

Notification Management:

The entire system is connected to few system generated emails that is achieved with the help of the architecture explained. Email Notifications are sent out whenever one of these event is triggered. The mail system is robust and real-time [2]. This helps in alerting mechanism to be implemented. When a new user is added, his details are passed through the mail. Alerts on changing the password is provided as well. The functionality of changing password whenever a user is unable to remember his/her password is also provided with the help of reset links [3, 9]. The admin also gets an option to alert a user when- ever he/she is dormant for a period of time and have not started the activity assigned to them

OAuth 2.0:

OAuth Server acts as the authentication and the authorization server for the entire application. The method to save the credentials in encrypted format was done with the help of BCrypt Encoder with 12 rounds of hashing. The tokens were not stored and were generated in-memory. The code generated at the beginning is a one-time code which improved the security further. SSL was enabled at the browser side which helped in prevention of network leakage and provide the secure layer of https. This entire flow was developed and implemented with the help of Spring Security and Spring OAuth.

6. Conclusion

This paper presented an overview on gamification based learning.

References

[1] R. M. Meloca, R. Re, and A. L. Schwerz, "An analysis of frameworks for microser- vices," in 2018 XLIV Latin American Computer Conference (CLEI), IEEE, Oct. 2018.
[2] N. N. J. Kreps and J. Rao., "Kafka: A distributed messaging system forlog pro- cessing," 2011.
[3] M. Argyriou, N. Dragoni, and A. Spognardi, "Security flows in OAuth 2.0 frame- work: A Case study," in Lecture Notes in Computer Science.
[4] V. Armenise, "Continuous delivery with Jenkins: Jenkins solutions to implement Continuous delivery," in 2015 IEEE/ACM 3rd International Workshop on Release Engineering, IEEE, May 2015.
[5] D. Merkel, Docker: lightweight Linux containers for consistent development and deployment. 2014.
[6] B. S. Scott Chacon, Pro Git. APRESS L.P., Dec. 24, 2014.
[7] Jenkins, "Jenkins user documentation," https://jenkins. Io/doc/
[8] D. Inc. "Docker documentation." https://docs.docker.com/...
[9] Okta, "Getting started - oauth," https:// oauth.net/getting-started/
[10] AWS, "AWS documentation."
[11] "Apache kafka Documentation," https: kafka.apache.org/documentation/...