# EASE AND SECURE LEAVE MANAGEMENT SYSTEM IN CLOUD COMPUTING ENVIRONMENT

**S. Rajakumar[1], A.S. Anakath[2], R. Kannadasan[3] & S. Ambika[4]**

Correspondence: anakatharasan@gmail.com

## ABSTRACT

Provisions on availing different categories of leaves are important in any type of organization in this world of information technology. Providing a system on the internet intended to solve the leave problems of the employees in an organization is inevitable now a days. This system assumes that the level of authority in organizations follows a tree structure. An employee is viewed as internal node of a tree in an organization and involves in two roles, one as a supervisor and another as a supervisee. The supervisee is viewed as leaf node of a tree. Some employees only are able to act as supervisee. Through this system the employee can apply leave to his respective supervisor i.e. to the parent node of an supervisee node. The supervisor is able to check and approve the leave of his supervisees. The major features include this system is independent of any operating syetem so that any computer system with a browser and internet connection is sufficient for the operation of this system, ease to use and afterall, enhanced security features. An employee is propely authenticated before allowed to enter into this system. There is an admin other than is responsible for adding and removing employees from an organization i.e. adding and removing nodes from a tree. This system is planned to be implemented in a cloud environment. The results show that this system is implemented and used in ease manner and enhanced authentication and approval of leave process is achieved.

**Keywords**: easy-leaves, leave system, MVC pattern, supervisor, supervisee

## 1. INTRODUCTION

Easy-leaves is a user-friendly online system. The main motto of Easy-leaves is to help employees to manage the leaves easily. The easy-leaves includes admin login and employee login. Once the employee is logged in, he can apply leave, view the leave status, cancel the leave, view leave history, view and manage the leaves of his supervisees, check the leave history of his supervisees. Once the admin of the organisation is logged in, he can add the new employee to the organisation. He can manage the employees in the organisation. The login system is same for both employee and admin. Only the users that are logged in can make changes to the data in the database. The interface is more user-friendly. The project has been developed from node.JS as server language and mongo DB as database.

## 2. LITERATURE SURVEY

In this work the authors use the idea of brainstorming to come up with an idea of managing leaves easily. The main aim of this idea is to develop an eazy leave management portal that is of important to an organization (Vikrant Kumar Kaushik et al., 2017). NoSQL stands for Not only SQL. These are the set of database management systems that are emerged to overcome the limitations of traditional relational database management systems. these Database systems are Schema-less. Which makes them much faster in doing the CRUD

[1] Dept. of Mathematics, University College of Engineering Ariyalur, TN, India.
[2] School of Computing, E.G.S.Pillay Engineering College, Nagapattinam, TN, India.
[3] Dept. of Software Systems, SCOPE, VIT University, Vellore, TN, India.
[4] Dept. of Computer Science and Engineering, University College of Engineering Ariyalur,TN, India.

operations (Ahmed et al., 2018). Node.js helped the Full Stack web developers to manage the server side and client side on their own. The applications developed in node.js are fast and highly reliable, due to its asynchronous approaches and non-blocking I/O(Shah et al., 2017).

Web GIS is based on node.js. It provides accurate and efficient special information to users. The performance of the web GIS server is also increased to offer facial information service (Chen et al., 2014). Front end developers will create the websites using HTML, CSS and JavaScript. Backend developers build web apps using Node.js, PHP,JSP<servlets etc. IOS and Android developers will build native mobile apps (Jun et al., 2013).

Node.js and PHP are server-side programming languages. Node.js use plain old JavaScript instead of introducing a new programming language. Node.js is gaining popularity and PHP is becoming obsolete (Bridge, 2016). [6]. Bootstrap is a frontend CSS and JavaScript library. Using bootstrap, one can easily add styling to the webpage to make it more responsive (Wales, 2014). This paper discusses the technologies used for development of web applications. The web app typically involves client and the server. The frameworks or the technologies used in client side are Angular, React.js, Vue.js etc. The frameworks or the technologies used in server side are Node.js, PHP, ASP.NET etc. (Mardan, 2013).

## 3. METHODOLOGY

The process model used for this work is V-shaped model. So, more emphasis is given to testing of components. The work strictly follows the MVC pattern. This system consists of the following components.

### 3.1 Classes

This component contains all the coding parts related to database. We create classes with appropriate attributes and operations in this component and the coding part that is necessary for creating and manipulating data in database is also done in this component. The employee class deals with storing and retrieving data related to employee from the database. Similarly, the admin and leave classes are also coded.

### 3.2 User Interfaces

This component contains all the coding parts required to make the user interfaces in the browser. This includes several sub componrnts like designing the starting page, navigation page that is common to whole webapp, interfaces used by employees, admin and interfaces used in authenticating users.

### 3.3 Server

This is component is responsible for running the server on the machine. Without this component all other components become useless. Whenever a request is sent by the user in the browser, the request first enters the server and then goes into all other required components to get the response. Usually the server component will be linked to the route’s component.

### 3.4 Routes

This is the component is responsible for handling different routes. Let's say user enters htttp://easy-leaves.herokuapp.com/admin, now this component will decide which controller sub component should be called to send a correct response to the user. Usually the routes component is linked to the controller's component. This component is further divided into routes related to employee, login, admin.

### 3.5 Controllers

This component actually does the real working part by relying on other components like classes and user interfaces. This component is the one that decides and responses the user, whenever he sends a request in a browser. The classes defined in that the class component are instantiated to get the real work done. This also links the interfaces designed in the user interfaces component to the response that is being sent to the user. The controllers are usually linked to classes and user interfaces.

### 3.6 Public

This component is tightly linked to user interfaces component. This contains all the coding part that are publicly available to the user. So, the user can actually see the code that is implemented in this component. Usually this module contains the code that is needed to enhance the user interfaces in user interfaces component.

### 3.7 Third Party Modules

These component are not implemented by the software project team but are used in the system development. For example, sending an email to the user. All these components are consolidated in Figure 1.

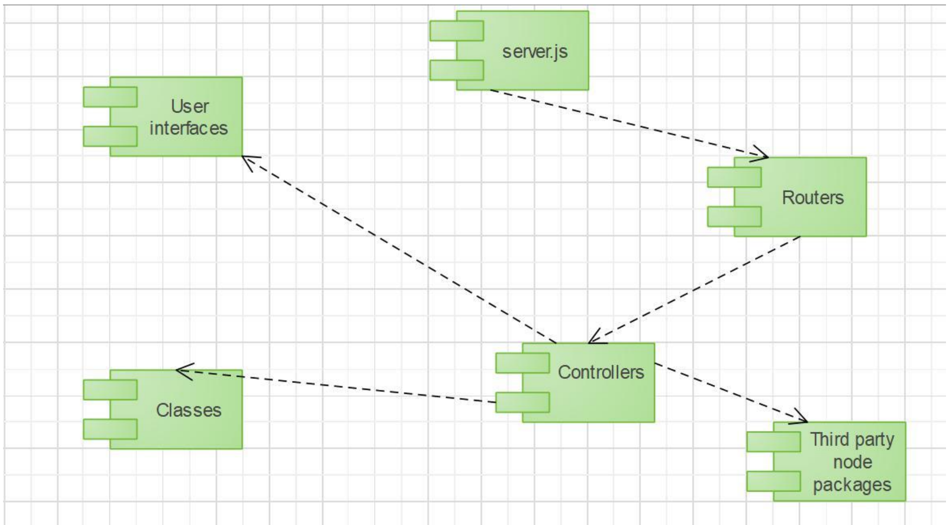


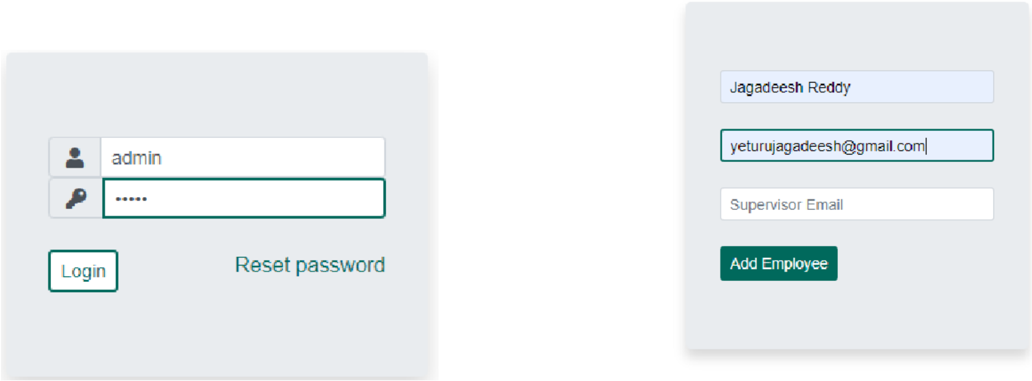
*Figure. 1.Block Diagram for Easy-leaves*

## 4. RESULTS AND DISCUSSION

### 4.1 Output

The Easy-leaves has been designed in the way that any level of “end user” will be managing to easily get familiarised with the interface of the system, within a short span of time.

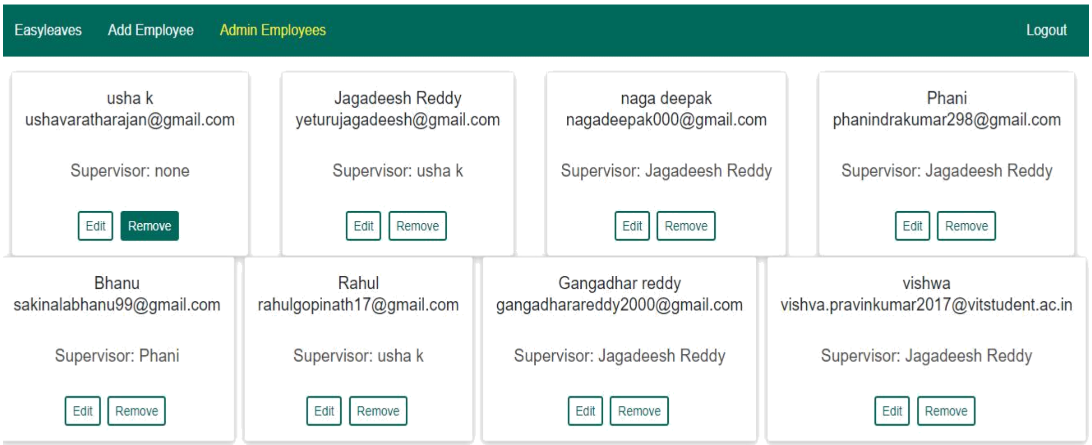
The admin logins into the web system.The admin is responsible for adding a new employee to the organisation. The admin adds the new employee to the organisation. The admin gives the employee’s name, email address and supervisor email. If the supervisor email field is left blank there won’t be any supervisor for that employee. These components are shown in Figure 2.


admin
•••••

Jagadeesh Reddy
yeturujagadeesh@gmail.com
Supervisor Email

*Figure. 2 Admin Login and Adding Employee*

The admin can manage all the employees in the organisation by clicking the admin employees link in the navigation bar. Now admin can remove or edit the details of an employee as shown in Figure 3.


usha k
ushavaratharajan@gmail.com
Supervisor: none

Jagadeesh Reddy
yeturujagadeesh@gmail.com
Supervisor: usha k

naga deepak
nagadeepak000@gmail.com
Supervisor: Jagadeesh Reddy

Phani
phanindrakumar298@gmail.com
Supervisor: Jagadeesh Reddy

Bhanu
sakinalabhanu99@gmail.com
Supervisor: Phani

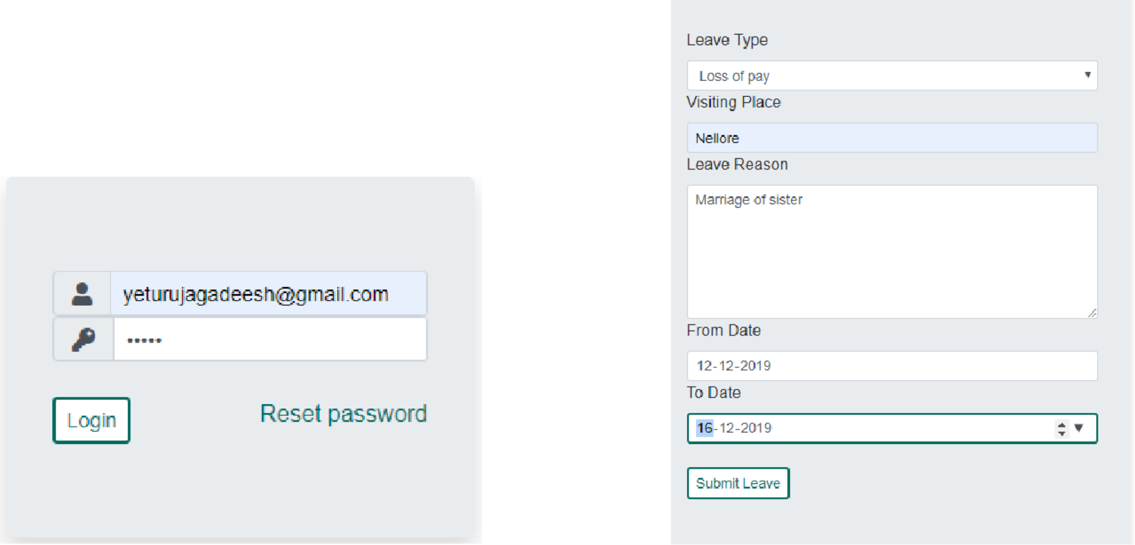
Rahul
rahulgopinath17@gmail.com
Supervisor: usha k

Gangadhar reddy
gangadharareddy2000@gmail.com
Supervisor: Jagadeesh Reddy

vishwa
vishva.pravinkumar2017@vitstudent.ac.in
Supervisor: Jagadeesh Reddy

*Figure. 3. Admin Employees*

Once the employee has been added to the organisation. The employee will get username and password as a mail. Now employee can login to the system using those credentials. Once user is logged in as an employee. Now he can apply the leave to his supervisor. This is shown in Figure 4.



yeturujagadeesh@gmail.com

•••••

Leave Type

Loss of pay

Visiting Place

Nellore

Leave Reason

Marriage of sister

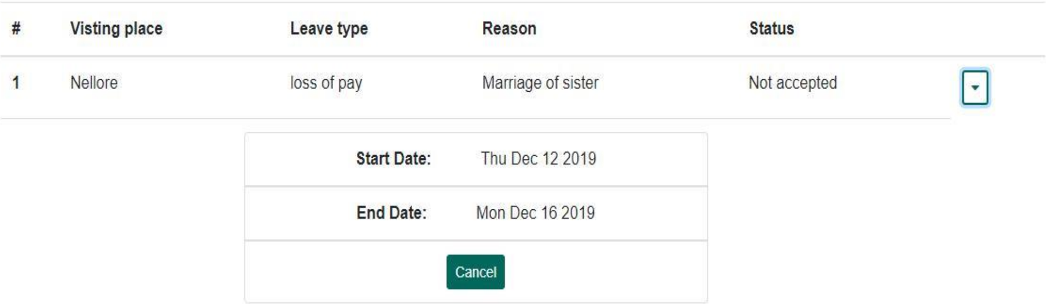
From Date

12-12-2019

To Date

16-12-2019

*Figure. 4. Employee Login and Applying Leave*

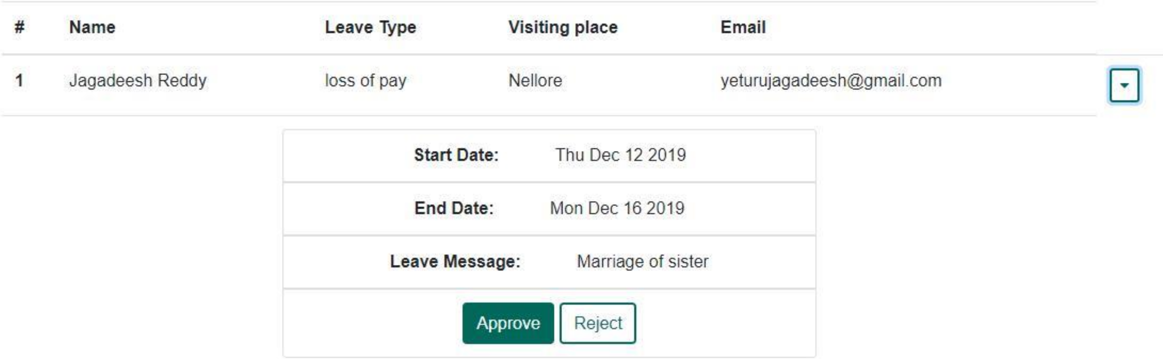


| # | Visting place | Leave type | Reason | Status |
|---|---|---|---|---|
| 1 | Nellore | loss of pay | Marriage of sister | Not accepted |

| | |
|---|---|
| Start Date: | Thu Dec 12 2019 |
| End Date: | Mon Dec 16 2019 |

*Figure. 5. Leave Status*

Once the leave has been applied, he can check the status using the leave status page. This is shown in Figure 5. Even he can cancel the leave if he wants to.

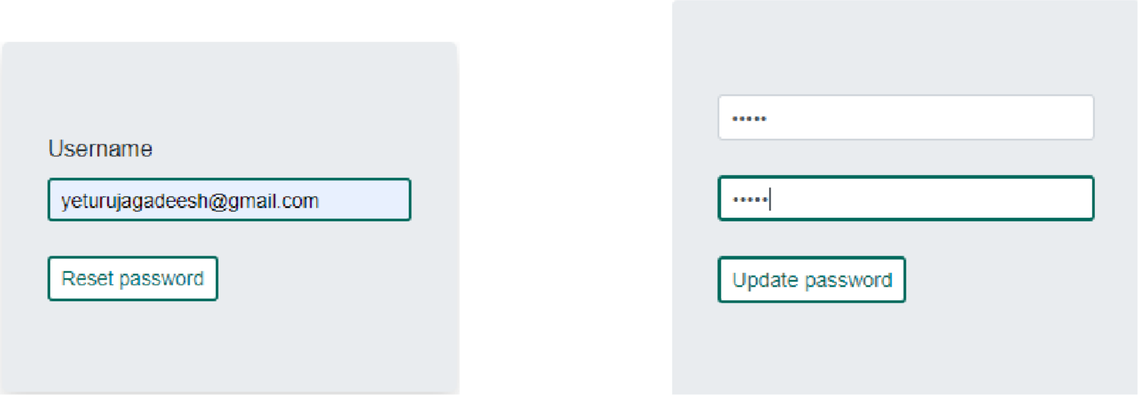


| # | Name | Leave Type | Visiting place | Email |
|---|---|---|---|---|
| 1 | Jagadeesh Reddy | loss of pay | Nellore | yeturujagadeesh@gmail.com |

| | |
|---|---|
| Start Date: | Thu Dec 12 2019 |
| End Date: | Mon Dec 16 2019 |
| Leave Message: | Marriage of sister |

*Figure. 6. Leave Received from the Supervisee*

Once the employee applies the leave, the supervisor can see the leaves of his supervisees in the leaves received page. He can approve or reject the leave. This situation is shown in Figure 6.



Username

yeturujagadeesh@gmail.com

*Figure. 7. Reset Password and New Password*

If the user is not happy with his password. He can request for change password in reset password page by entering his username as shown in Figure 7. After submitting the reset password request, the user gets an email with the link to set the new password.

### 4.2 Test Cases

Variety of test cases are generated for implementing the testing strategies in our system. Table 1 shows test cases on login, Table 2 shows the test cases on add employee and Table 3 shows test cases on applying leave.

*Table1. Test Cases on Login*

| Test Scenario Id | Test Scenario Description | Test Case Id | Test case Description | Test Steps |
|---|---|---|---|---|
| TS_001 | Verify the login functionality | TC_Login_001 | Enter the username & valid password | 1. Enter valid Username<br>2. Enter valid Password<br>3. Press on login button |
| TS_001 | Verify the login functionality | TC_Login_002 | Enter a valid username & invalid password | 1. Enter valid loginid<br>2. Enter invalid password<br>3. Press on login button |
| TS_001 | Verify the login functionality | TC_Login_003 | Enter a invalid username & valid password | 1. Enter invalid loginid<br>2. Enter valid password<br>3. Press on login button |
| TS_001 | Verify the login functionality | TC_Login_004 | Enter a invalid username & invalid password | 1. Enter invalid loginid<br>2. Enter invalid password<br>3. Press on login button |

*Table2: Test Cases on Add employee*

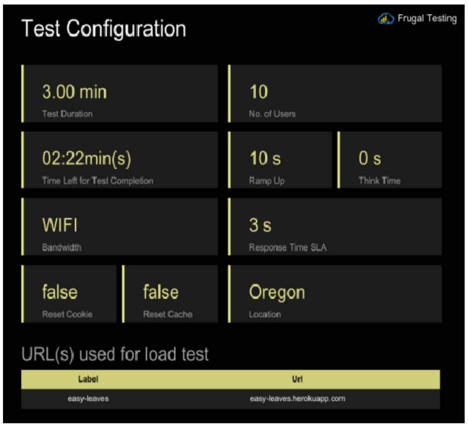
| Test Scenario Id | Test Scenario Description | Test Case Id | Test case Description | Test Steps |
|---|---|---|---|---|
| TS_002 | Verify the add employee functionality | TC_Add_001 | Enter valid employee email & valid supervisor email | 1. Enter employee name<br>2. Enter valid employee email<br>3. Enter valid supervisor email |
| TS_002 | Verify the add employee functionality | TC_Add_002 | Enter valid employee email & invalid supervisor email | 1. Enter employee name<br>2. Enter valid employee email<br>3. Enter invalid supervisor email |
| TS_002 | Verify the add employee functionality | TC_Add_003 | Enter invalid employee email & valid supervisor email | 1. Enter employee name<br>2. Enter invalid employee email<br>3. Enter valid supervisor email |
| TS_002 | Verify the add employee functionality | TC_Add_004 | Enter invalid employee email & invalid supervisor email | 1. Enter employee name<br>2. Enter invalid employee email<br>3. Enter invalid supervisor email |

*Table3: Test Cases on Applying Leave*

| Test Scenario Id | Test Scenario Description | Test Case Id | Test case Description | Test Steps |
|---|---|---|---|---|
| TS_003 | Verifying the applying leave functionality | TC_Leave_001 | Enter a valid start date and a valid end date | 1. Enter leave type<br>2. Enter visiting place<br>3. Enter leave message<br>4. Start date<br>5. End date |
| TS_003 | Verifying the applying leave functionality | TC_Leave_002 | Enter a valid start date and a invalid end date | 1. Enter leave type<br>2. Enter visiting place<br>3. Enter leave message<br>4. Start date<br>5. Invalid end date |
| TS_003 | Verifying the applying leave functionality | TC_Leave_003 | Enter a invalid start date and a valid end date | 1. Enter leave type<br>2. Enter visiting place<br>3. Enter leave message<br>4. Invalid start date<br>5. End date |
| TS_003 | Verifying the applying leave functionality | TC_Leave_004 | Enter a valid start date and a valid end date | 1. Enter leave type<br>2. Enter visiting place<br>3. Enter leave message<br>4. Invalid start date<br>5. Invalid end date |

## 4.3 Testing Tool Performance

The test tool used is fugaltesting.com are used in this system. the configuration of the testing procedure applied on the proposed system includes the duration of the test, the number of users, the response time along with other relevant configuration features. An overall summary of the performance of the system provides theaverage throughput, percentage of error in the system and average response time. Graphical interpretation of Bandwidth, Average Throughput and Average Response Time with respect to the time has also been depicted in Figure 8.



Test Configuration

Frugal Testing

3.00 min
Test Duration

10
No. of Users

02:22min(s)
Time Left for Test Completion

10 s
Ramp Up

0 s
Think Time

WIFI
Bandwidth

3 s
Response Time SLA

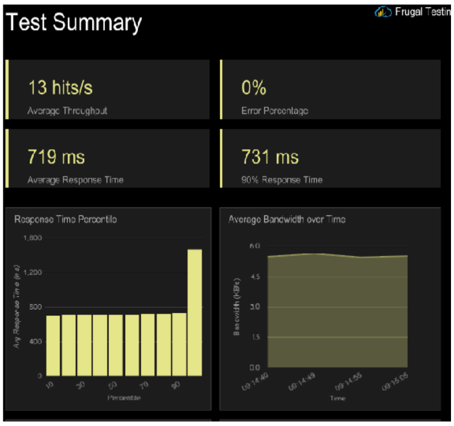
false
Reset Cookie

false
Reset Cache

Oregon
Location

URL(s) used for load test

| Label | Url |
|---|---|
| easy-leaves | easy-leaves.herokuapp.com |



Test Summary

Frugal Testin

13 hits/s
Average Throughput

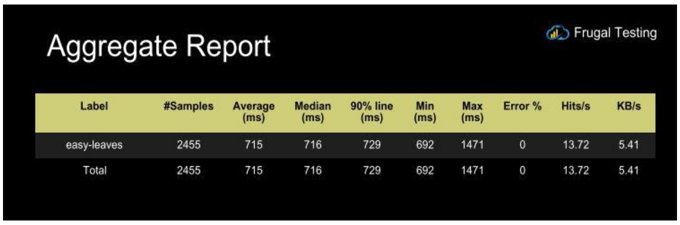
0%
Error Percentage

719 ms
Average Response Time

731 ms
90% Response Time

*Figure. 7. Test configuration Test Summary*

A tabular representation of the aggregate report, which includesthe error percentage, the number of hits per second as well the number of samples that have been tested is shown in Figure. 8.



Aggregate Report

Frugal Testing

| Label | #Samples | Average (ms) | Median (ms) | 90% line (ms) | Min (ms) | Max (ms) | Error % | Hits/s | KB/s |
|---|---|---|---|---|---|---|---|---|---|
| easy-leaves | 2455 | 715 | 716 | 729 | 692 | 1471 | 0 | 13.72 | 5.41 |
| Total | 2455 | 715 | 716 | 729 | 692 | 1471 | 0 | 13.72 | 5.41 |

*Figure. 8. Aggregate Report*

## 5. CONCLUSION

In the case of leave management systems, the process of applying leave and other functionalities offered should be essentially streamlined, without human interaction with it to compete in the external market. In this system, we have developed an automated leave management system using node.js, mongoDB, HTML and CSS. The proposed prototype application is authenticated in accomplishing security of the system and has been tested by URL test, using frugal testing CASE tool.

## 6. REFERENCES

1. Vikrant Kumar Kaushik, Arjun Kumar Gupta, Ashish Kumar and Abhishek Prasad (2017). Student leave Management System. *IJARIIE*, 3(5), 1-5.
2. Ahmed, Md. Razu, Ali, Md. Asraf and Sundaraj, Kenneth (2018). A literature review on NoSQL database for big data processing. *International Journal of Engineering & Technology,* 7(2), 2344-2348.
3. Shah, H., Hezbullah, S., and Soomro, T. (2017). Node.js Challenges in Implementation. *Global Journals Inc. (USA),* 17(2), 234-239.
4. Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey, in Mobile Networks and Applications. *Springer-Verlag New York, Inc*, 19(2), 749-756.
5. Jun, S.H., and Doh, K.T. (2013). Design and Implementation of Web GIS Server Using Node.js. *Journal of Korea Spatial Information Society*,21(3), 654-661.
6. Bridge, F. (2016). *What Types of Developers Are There?.* tree house.
7. Wales, M. (2014). *Front-End vs Back-End vs Full Stack Developers*. Udacity.
8. Mardan, A. (2013). *PHP vs. Node.js.* Programming Weblog.
9. Safwana (2017). *Why Should You Use Bootstrap*.
10. Mandava, A., and Antony, S. (2012). *A review and analysis of technologies for developing web applications.*