**SYNOPSIS**

**Report on**

**STUDENT REPOSITORY SYSTEM**

**by**

Akash Yadav 2000290140013

Vyom Gupta 2000290140137

**Session:2021-2022 (4th Semester)**

Under the supervision of

**Ms. Divya Singhal (Assistant Professor)**

### KIET Group of Institutions, Delhi-NCR, Ghaziabad



### Department Of Computer Applications

**KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206**

(march- 2022)

**ABSTRACT**

This project is insight into the design and implementation of a Student Repository System. Hence to solve this issue we have come up with a java-based web application where all the students & teachers & their parents can stay connected together on a single platform, thereby helping all the students to be in constant touch with the faculties and making it possible for them to get their problems solved in an instant. Hence, this web-based application will help students to keep track of their performances & marks. And most important their parents can track their child anytime just with an active internet connection.

**TABLE OF CONTENTS**

1. Introduction 2
2. Literature Review
   1. Introduction
   2. Historical Overview
      1. Instant Messaging: Child Of The 90’s 3
3. Project Objective 4
4. Research Methodology 4
5. Project Outcome 5
6. Proposed Time Duration 4

References 6

1. **INTRODUCTION**

Managing a single student’s information is very complex and we know that a college have a huge amount of students’ data so managing it that too manually is a very challenging task for any institute. Also, now that we are going with a battle with the covid pandemic, so the management has become more difficult to do.

Also, it has become very difficult for the college management to convey any important notice or message to all the students as all the college faculty members & their parents. Hence to solve this issue we have come up with an java-based web application where all the students & teachers & their parents can stay connected together on a single platform, thereby helping all the students to be in constant touch with the faculties and making it possible for them to get their problems solved in an instant. And the parents can also check all the progress of their child.

Hence, this web-based application will help students to be in contact with their performance. And without depending upon any third-party software for communication.

**2. LITERATURE REVIEW**

**2.1 Introduction**

Web apps now have more global users than traditional standalone apps—which means that they can be accessed anyone with having internet. Today only to kinds of applications are there, either web based or mobile based. Standalone apps are not in much use today as all the of the world has moved over to the internet. So having our application on the web or mobile is the best way to provide a solution to anyone.

**2.2 Historical Overview**

**2.2.1 Manual System:**

Initially the management of student in any institute would happen on the manual registers only. This gave room to many problems to the student and teachers and staff of that institute also. Any file or paper could misplace or lost or stolen. No real time view facility was there. If any student or teacher need to check or make any changes in the data then they have to present on the location. They cannot access the information remotely.

**2.2.2. Digital System:**

Then came the computers and they changed the whole scenario. Now student can check real time performance. Teachers can handle the data from anywhere they want. Also, parents can check the performance of their child. All because of the internet. And thus web apps or mobile apps

1. **PROJECT OBJECTIVE**

**3.1 Staff Managing Things**

One of the fundamental tasks is to give the ease to all the teachers and department to manage the student’s information. Be it their marks, their performance, their assignments, or placements activities. All other things like fee payment, any late dues, fines etc can be managed easily.

**3.2** **Student Access**

By giving a tool to a student they can also check their performance, and marks, and attendance through a single app. No need to worry about going to premises every time. Also uploading the documents become easy. They can submit their fee from anywhere anytime. Any fault can also be track easily.

**3.3 Parent Access**

By giving a medium for parents that they can check and keep an eye on their child from their home without coming to the institute is very helpful. They can access all the information of their child by just a single web or mobile app.

1. **PROJECT OUTCOME**

The project successfully delivered on all requirement specification specified by the user. Care was ensured during the design to make sure data integrity is maintained and to avoid all forms of redundancies associated with data.

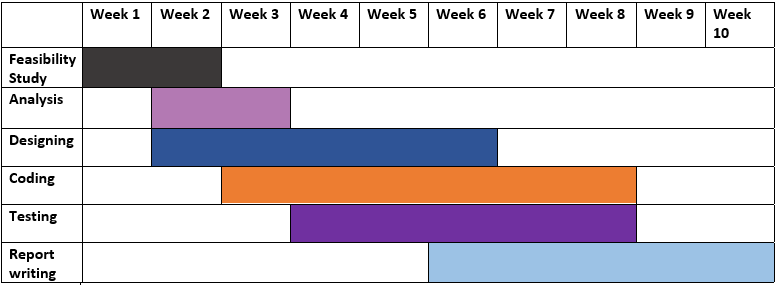
The user is assured a very friendly interface, behind which there are wide ranging technical details that went in. The user guide is a mere formality because, the project was specially created bearing in mind interaction and designs that would make users feel as though they have used a system such as this.

This project has also been built in such a manner that future changes or modifications that are required can easily be implemented without affecting the functionality of the system. This project is used on android environment, and can be used on any version so it can be used by individuals with different levels of android devices.

The technical document that is provided in the synopsis of this project will help developers understand the internal workings of the system.

1. **Proposed Time Duration**

**Gantt chart**

****

**Feasibility Study**: we have decided 2 weeks for feasibility study and requirement gathering for that we can analysis our capabilities and resources.

**Analysis:** form 2nd to 3rd week we proposed our self for esteem analysis of software requirements and risk and resource management.

**Designing:** In this phase (from 2nd week to 6th week) we will focus on designing the blueprint of software and tries to focus on coding part also.

**Coding:** Form week 3rd to week 8, we focus on coding part and tries to follow pre developed prototype of software.

**Testing:** Testing is not a part of only testing phase hence testing will be applied through each and every phase of software development life cycle.

**Report Writing:** During the process of developing project (software) we will constantly writes report on current project**.**

**\*Hence the total time required to develop this project is around 10 weeks.**

**REFERENCES**

1. Aaron, P., (2020) “Webrtc vs websockets.” http://stackoverflow.com/a/18825175, sep 2013. Accessed: 2017-04-19.
2. Ackley, B., (2020) “Webrtc samples.” https://webrtc.github.io/samples/, sep 2013. Accessed: 2017-04-19.
3. Abby, T., (2020) “User stories: An agile introduction.” http://www.agilemodeling.com/ artifacts/userStory.htm. Accessed: 2016-10-20.
4. Bairam, W., (2020) “Making the switch from making the switch from node.js to golang.” http://blog.digg.com/post/141552444676/ making-the-switch-from-nodejs to-golang, mar 2016. Accessed: 2016-10- 19.
5. Callum, S., (2020) “Sinon - best practices for spies, stubs and mocks.” https://semaphoreci.com/community/tutorials/best-practices-for-spies-stubsandmocks-in-sinon-js, 2016. Accessed: 2016-10-03.
6. Caleb, P., (2020) “Why is nosql faster than sql?.” http://softwareengineering. stackexchange.com/questions/175542/why-is-nosql-faster-than-sql, nov 2012. Accessed: 2016-10-21.
7. Dakari, U., (2020) “Why nosql.” http://softwareengineering.stackexchange.com/ questions/175542/why-is-nosql-faster-than-sql. Accessed: 2016- 10-21.
8. Dashin, O., & Shanda, R., (2020) “Exploring the different types of nosql databases.” https://www. 3pillarglobal.com/insights/exploring-the-different-typesof-nosql-databases, may 2015. Accessed: 2017-04-29.
9. Daen, I., (2020) “Composition vs inheritance - react.” https://facebook.github.io/react/ docs/composition-vs-inheritance.html. Accessed: 2016-12-18.
10. D. T. Andrew Hunt, The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley Professional, first ed., october 1999.
11. Faryn, F., (2020) “Mongodb manual 3.4.” https://docs.mongodb.com/manual/reference/ limits/. Accessed: 2017-08-30. 34
12. Gandale, A., (2020) “6 rules of thumb for mongodb schema design.” http://blog.mongodb.org/post/87200945828/ 6-rules-of-thumb-for-mongodbschema-design-part-1, may 2014. Accessed: 2016-10-18.
13. Hakam, T., (2020) “Scaling secret: Real-time chat.” https://medium.com/always-becoding/ scaling-secret-real-time-chat-d8589f8f0c9b#.m5jigxq6x, may 2015. Accessed: 2016-10-18.
14. Hale, B., (2020) “Analysis of json use cases.” https://blogs.oracle.com/xmlorb/entry/ analysis\_of\_json\_use\_cases, apr 2013. Accessed: 2016-11-08.
15. Icharus, H.,“Crud cycle (create, read, update and delete cycle).” http:// searchdatamanagement.techtarget.com/definition/CRUD-cycle. Accessed: 2016-11- 08.
16. Idys, E., & Sam, D.,(2020) “About native xmlhttp.” https://msdn.microsoft.com/enus/library/ ms537505(v=vs.85).aspx. Accessed: 2016-12-18.
17. Kaija, S.,(2020) “Please. don’t patch like an idiot..” http://williamdurand.fr/2014/02/14/ please-do-not-patch-like-an-idiot/, aug 2016. Accessed: 2016-12-18.
18. K.D, & Ernest, A.,(2020)“Rfc 5789 - patch method for http.” https://tools.ietf.org/html/rfc5789, mar 2010. Accessed: 2016-12-18.
19. Lang, Y.,(2020)“Perfomance tips | google cloud platform.” https://cloud.google.com/ storage/docs/json\_api/v1/how-tos/performance#patch. Accessed: 2016- 12-18.
20. Najib, M.,(2020)“container vs component?.” https://github.com/reactjs/redux/issues/ 756#issuecomment-141683834, sep 2015. Accessed: 2016-11-13.
21. Oisin, H., (2020) “Html5 websocket: A quantum leap in scalability for the web.” http://www. websocket.org/quantum.html. Accessed: 2016-11-13.
22. Sam, T., (2020) “Building a scalable node.js express app.” https://medium.com/@zurfyx/ building-a-scalable-node-js-express-app1be1a7134cfd, feb 2017. Accessed: 2017-04-13.
23. Sadhbh, I., “Strategy design pattern.” https://sourcemaking.com/design\_patterns/ strategy. Accessed: 2016-11-14. 35 24. T’Marius, A., (2020) “Dynamic testing.” https://www.tutorialspoint.com/software\_testing\_ dictionary/dynamic\_testing.htm, apr 2017. Accessed: 2017-04-03