

# **Time Management and Optimization Technique for Efficient Project Management and Completion**

*Submitted in partial fulfillment of the requirements for the degree of*

**Bachelor of Technology**

in

**Computer Science and Engineering**

*by*

**Nithish S**

**17BCE0002**

**S Akshay Prassanna**

**17BCE0041**

**K Nithiya Soundari**

**17BCE2244**

**Under the guidance of**

**Dr. MARGRET ANOUNCIA S**

**School of Computer Science and Engineering**

**VIT, Vellore.**



June, 2021

## **DECLARATION**

I hereby declare that the thesis entitled “Time Management and Optimization Technique for Efficient Project Management and Completion” submitted by me, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Dr. Margret Anouncia S.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

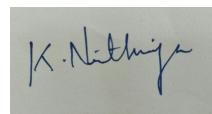


Place: Vellore

Date: 5/6/2021

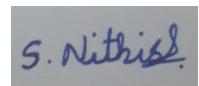
Akshay Prassanna S

17BCE0041



Kirubakaran Nithiya Soundari

17BCE2244



S Nithish

17BCE0002

**Signature of the Candidates**

## **CERTIFICATE**

This is to certify the thesis entitled “Time Management and Optimization Technique for Efficient Project Management and Completion” submitted by **Nithish S (17BCE0002)**, **S Akshay Prassanna (17BCE0041)** & **Kirubakara Nithiya Soundari (17BCE2244)**, SCOPE School, VIT, for the award of the degree of *Bachelor of Technology in Computer Science and Engineering*, is a record of bonafide work carried out by him/her under my supervision during the period, 01. 12. 2020 to 09.06.2021, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion, meets the necessary standards for submission.

Place : Vellore

Date : 5/6/2021

**Signature of Guide**

**Internal Examiner**

**External Examiner**

**Head of the Department**

**B.Tech Computer Science and Engineering**

**SCOPE School**

## **ACKNOWLEDGMENTS**

Firstly, I would like to express my heartfelt gratitude to Dr. Margret Anouncia S for being kind enough to let me work under her guidance, the SCOPE school , the Dean Dr. Ramesh Babu K and the SCOPE Head of Department Dr. Vairamathu S for allowing me to work on this project. I would also like to thank them for diligently teaching and training me, despite their busy schedules, about the various proceedings pertaining to the project thereby enhancing our theoretical and practical knowledge. I would also like to thank my fellow peers for supporting me when it was needed. Last but not the least, I would like to thank Chancellor Dr. G. Viswanathan, the Vice-Chancellor Dr. Ram Babu Kodali, the Pro ViceChancellor Dr. S. Narayanan and the Vellore Institute of Technology for providing me the environment to work, gain knowledge and grow.

Nithish S (17BCE0002)

S Akshay Prassanna (17BCE0041)

Kirubakaran Nithiya Soundari (17BE2244)

## **EXECUTIVE SUMMARY**

Projects have become a very important aspect in any student's academics or any employee's career. It is becoming more and more challenging for anyone to keep track and be on track with all the work they have to complete. There are a lot of tools out there to help a person with his/her productivity on a daily basis. Let it be a student or an adult, projects are playing such an important role in their career aspect and it is very hard to meet deadlines if there is no proper planning.

The most popular algorithms for project management are the Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM). A detailed analysis of these algorithms combined with good implementation can bring out a very useful tool that can be used by anyone to manage their group or even individual projects, making it easy to use and understand the statistics and details generated by the tool.

Now, this tool incorporated with a collaboration portal where people all around the world can meet, discuss, create new projects will be helpful for a lot of individuals looking forward to improving their skills and putting them to practical use. Hence, the main aim of the project is to create a collaboration portal/platform where people can meet, collaborate and do projects while expanding their network. The PERT and CPM algorithms will be implemented using Javascript, the front-end of the website will be done using Reactjs, the backend using nodejs and the backend database will be MongoDB which uses mongoose to talk to the backend of the application.

**Keywords:** Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM), Javascript, Reactjs, Nodejs, MongoDB, Mongoose

<b>CONTENTS</b>	<b>Page No.</b>
<b>Acknowledgement</b>	I
<b>Executive Summary</b>	II
<b>Table of Contents</b>	III
<b>List of Figures</b>	VI
<b>List of Tables</b>	VII
<b>Abbreviations</b>	VIII
<b>1. INTRODUCTION</b>	
1.1. Theoretical Background	1
1.2. Motivation	2
1.3. Aim of the Proposed Work	3
1.4. Objective(s) of the Proposed Work	4
<b>2. Literature Survey</b>	
2.1. Survey of the Existing Models/Work	5
2.2. Summary/Gaps identified in the Survey	5
<b>3. Overview of the Proposed System</b>	
3.1. Introduction and Related Concepts	8
3.2. Framework, Architecture or Module for the Proposed System(with explanation)	13
3.3. Proposed System Model(ER Diagram/UML Diagram/Mathematical Modeling	21
<b>4. Proposed System Analysis and Design</b>	
4.1. Introduction	23
4.2. Requirement Analysis	
4.2.1. Functional Requirements	
4.2.1.1. Product Perspective	23
4.2.1.2. Product Features	24
4.2.1.3. User Characteristics	25
4.2.1.4. Assumptions & Dependencies	26
4.2.1.5. Domain Requirements	26
4.2.1.6. User Requirements	28

4.2.2.	Non Functional Requirements	
4.2.2.1.	Product Requirements	
4.2.2.1.1.	Efficiency (in terms of Time and Space)	29
4.2.2.1.2.	Reliability	30
4.2.2.1.3.	Portability	31
4.2.2.1.4.	Usability	31
4.2.2.2.	Organizational Requirements	
4.2.2.2.1.	Implementation Requirements (in terms of deployment)	32
4.2.2.2.2.	Engineering Standard Requirements	33
4.2.2.3.	Operational Requirements (Explain the applicability for your work w.r.to the following operational requirement(s))	33
4.2.3.	System Requirements	
4.2.3.1.	H/W Requirements(details about Application Specific Hardware)	36
4.2.3.2.	S/W Requirements(details about Application Specific Software)	36
<b>5.</b>	<b>Results and Discussion</b>	
5.1.	Project Dashboard	
5.1.1.	Team information	37
5.1.2.	Project Statistics	38
5.1.3.	Add Task	39
5.1.4.	Network Diagram	39
5.1.5.	All Tasks	40
5.1.6.	Group Chats	40
5.1.7.	Posts	41
5.2.	Dealing with projects	
5.2.1.	Create project	41
5.2.2.	Join project	42
5.2.3.	My projects	43
5.3.	The Networking Platform	43
5.4.	Chats	
5.4.1.	Basic working of Individual chats	48
5.4.2.	Basic working of Group chat	49
5.4.3.	Chat Block	50
<b>6.</b>	<b>Future work</b>	51
<b>7.</b>	<b>References</b>	54

**APPENDIX**

<b>APPENDIX A</b>	55
<b>APPENDIX B</b>	60
<b>APPENDIX C</b>	63

## LIST OF FIGURES

<b>Figure No</b>	<b>Title</b>	<b>Page No</b>
1.1.1	The phases of project management	2
1.2.1	Main reasons why projects fail	3
3.1.1	Solutions to avoid project failure	8
3.1.2	PERT Formula	10
3.1.3	CPM Formula	11
3.1.4	PERT Calculation Example	11
3.1.5	Mapping	12
3.2.1	MERN Stack Framework Architecture	14
3.2.2	Project management tool integration	14
3.2.3	Project Management Tool Functionality	17
3.2.4	Workflow of image moderation	18
3.2.5	Tensorflow and resnet-50	19
3.2.6	Workflow of Video Moderation	21
3.3.1	Entity Relationship Diagram	21
3.3.2	PERT CPM Component Integration	22
3.3.3	All Modules Connected Overview	22
5.1.1	Project Dashboard	37
5.1.1.1	Project - Team Info	38
5.1.2.1	Project Statistics	38
5.1.3.1	Creation of new tasks	39
5.1.4.1	Project Network Diagram	39
5.1.5.1	Project Tasks Board	40
5.1.6.1	Project Group Chat	40

5.1.7.1	Post tagged by Project	41
5.2.1.1	Creation of project	42
5.2.2.1	Joining of project	42
5.2.3.1	All projects of user	43
5.3.1	User profile - Personal information	44
5.3.2	Edit Profile	44
5.3.3	List of all users	45
5.3.4	Followers' data stored in DB	45
5.3.5	Following and Followers flow diagram	45
5.3.6	Following notification	46
5.3.7	Home page	46
5.3.8	Text post	46
5.3.9	Post data stored in DB	47
5.3.10	Alert notification on report of post	47
5.4.1.1	Follow users	48
5.4.1.2	Chat between 2 users	49
5.4.2.1	Group Chat	49
5.4.3.1	Block Users	50

## LIST OF TABLES

<b>Table No</b>	<b>Title</b>	<b>Page No</b>
2	Literature Survey	5
3.2.1	Difference between PERT and CPM	15

## **LIST OF ABBREVIATIONS**

1. PERT : Project (or Program) Evaluation and Review Technique.
2. CPM : Critical Path Method.
3. JS : Javascript
4. DB : Database
5. HTML : Hyper Text Markup Language
6. CSS : Cascading Style Sheet

## **1. Introduction**

### **1.1. Theoretical Background**

Everyone has experienced a big change over the past two years due to the COVID-19 pandemic. It has become the “new” normal to satisfy all our needs through online services. The world is going towards digitalization at a very fast pace after COVID-19 turned out to be a deadly virus. So many day-to-day activities such as buying groceries, exercise, going to school or work have all been shifted to digital media. Physically meeting people has very much reduced over the last year. There are many people of various age groups affected by this “new” normal. But, as many say, the most affected individuals are the students, actively pursuing their education. As all the classes have been transformed to online (E-learning), a student does not even get to know his / her classmates. And, no one has the answer to when will life go back to the way it was before this pandemic.

Even though there have been enormous changes due to the pandemic, the world is not going to stop till everything gets back to normal. Everyone still has to grow, excel and succeed in any condition. New technologies are being found, scientific advancements being made even in the “new” normal.

Almost all curriculums today have a component called projects. This was introduced as projects have always allowed individuals to practice and improve their skills in an application aspect that is suitable for the real world. There are five main phases for a project to be completed successfully and they are known as the phases of project management.

Normally students do their projects in groups while having several face-to-face meetings on how to proceed and complete the project. Due to the pandemic, this is not possible and hence a tool that can help with efficient project planning, scheduling and allocation of tasks will be useful for individuals to plan, execute and complete projects on time while working remotely. The project aims to build such a tool with a user-friendly user interface.



Figure 1.1.1 The phases of project management

## 1.2. Motivation

Projects have become an important aspect of any student's education and career. Let it be job recruiters or the admission panel, rather than seeing the scores and knowledge gained by the student, the projects and experiences gained by the students are looked upon more and given a higher weightage. Hence, it is clear that from a career point of view, projects play an important role.

Due to the very challenging and competitive environment, students tend to think more and come up with different ideas. One problem students face a lot is finding the right peers to work with. Excluding academic projects, there are so many individuals who spend time developing products and doing projects on their own. Many times, it is very hard to find people with the right skill set. That too networking being reduced for all students due to E-learning, they do not even get to interact with classmates about their skills. To solve this problem, a portal where any individual of any age, any experience can come collaborate with other people to do a project will be a great platform to meet more people, grow their network and improve one's skills.

The main motivation which led to the idea of creating a tool that will greatly help project management was the realization of how important assets such as time, resources and cost, when not used correctly without proper planning can lead to disastrous project management which in turn leads to the failure of the project. The diagram below depicts the main factors that lead to

project failure and this project will be aiming to come up with the solutions to the below-stated problems.



*Figure 1.2.1 Main reasons why projects fail*

### 1.3. Aim of the Proposed Work

The aim of the project has been formally stated here:

“A fully working project management system which consists of time, task visualization and management tools that make project planning, designing, scheduling and completion serene is the primary aim of the project. This system will additionally offer a networking platform which will enable users to find the right people required for their project.” Like-minded people with the same goals always bring success.

The project will be addressing the problems depicted in *Figure 1.1* by providing solutions that will be user-friendly and be efficient. Taking into note that there are already tons of project management software available, the tool developed in this project hopes to bring together all the important features needed for a student, in one single place, allowing him/her to go on with their projects on the fly from anywhere anytime.

#### 1.4. Objective(s) of the proposed work

A fully working collaboration portal with:

- ❖ Time management tool designed and developed using the PERT algorithm
- ❖ Task management tool which predicts the tasks with the level of priority designed and developed using the CPM algorithm
- ❖ An efficient task board to manage the status of tasks (todo, ongoing, review, completed) to enable easy task allocation and management.
- ❖ Group chats feature for the project team to discuss, plan and execute the project with high coordination and good communication.
- ❖ Posts for the entire community to get to know the different projects going on in the portal
- ❖ For users to expand their network and meet like-minded people online

## 2. Literature Survey

*Table 2.1. Literature Survey*

Name of Journal	Survey of the Existing Models/Work	Summary/Gaps identified in the Survey
<u>Time and Cost Optimization in Feasibility Test of CCTV Project using CPM and PERT</u> <i>Muhammad Ridho Bintang, Kelly Rossa Sungkono, Riyanto Sarno</i> <i>2019 International Conference on Information and Communications Technology (ICOIACT)</i>	<p>This paper talks about a research that aims to find the trade-off between cost and optimal time required to complete the project of CCTV by using the <i>Critical Path Method(CPM)</i> and <i>Project Evaluation and Review Technique(PERT)</i><sup>[1]</sup>. The research used the data about the project time and expenses from computer vision (CV) Aditya IT Consultant, a software house based in Kediri, East Java. The expected duration is 73 days, and the normal cost is IDR 207,858,300.</p>	<p>By using CPM and PERT they were able to reduce the working time by 25 days and the cost up to INR 39,250,000. The opportunity for the CCTV project to be completed is 99.71%, so the predicted project time by using PERT, which is 48 days, is acceptable.</p>
<u>Scheduling of House Development Projects with CPM and PERT Method for Time Efficiency</u> <i>Muhammad Kholil , Bonitasari Nurul Alfa , Madjumsyah Hariadi</i> <i>IConCEES 2017 - IOP Conf Series: Earth and Environmental Science</i>	<p>This paper talks about a research that aims to find an optimal completion time required for the construction of a house that has an area of 36 m<sup>2</sup>. The primary reason for project scheduling is to ensure that the deadline can be achieved. The use of both PERT and CPM methods in the scheduling of the house construction project has a very significant effect on the project completion time. The actual completion time</p>	<p>The total duration required for a home construction project using the CPM method is 131 days, or save time up to 42 days and by using the PERT method is 136 days. And the probability of completing the project is 74.54 %. This means that the possibility of home construction project activities can be completed on time is high enough. The difference between CPM and PERT calculation is 5 days. This is</p>

	required is 173 days.	because the CPM method has a definite duration estimation, whereas the PERT method has thereby estimation estimates.
<p><u>Optimizing Production Time in Book Printing using PERT/CPM</u></p> <p><i>Endang Yuniarti and Emmidia Djonaedi</i></p> <p><i>ASAIS 2019 - Annual Southeast Asian International Seminar</i></p>	This paper talks about a research that aims to produce an optimal production time with a case study of the production of a printed book entitled Brilliant which was produced for 20,000 units. The size of the book is 17.6 x 25 cm, Cover is Art Paper 210 GSM, the Content page is HVS 70 gsm, Color processes using 4 /0 color to cover and 1 /1 color to book; Number of pages is 115 pages; Finishing is with glue. Overall, the process includes three stages: pre-press, press and post press. The actual total production time needed is 45.2 hour or 2708 minutes.	The main purpose of this research is to obtain an Optimization Production Time in Book Printing using PERT / CPM approaches. After applying the PERT/CPM method the total production time needed is 2500 minutes or 41.5 hours.
<p><u>A Practical Approach using CPM/PERT for Certain Activity Times in Construction Parking Project</u></p> <p><i>Aznida Hayati Zakaria, Norizah Haron, Wan Suryani Wan Awang, Zarina Mohamad,Fadhilah Ahmad, Nurnadiyah Zamri, Ahmad Nazari Mohd Rose and Mustafa Mat Deris</i></p> <p><i>IOP Conference Series: Earth</i></p>	This paper checks into the construction line, where the most essential factors would be time, cost and project performance. CPM and PERT are presented as an efficient solution to the preparation and execution of these construction projects. In this paper, they have reviewed by the help of multiple SME contractors who were involved in the	The paper also suggests a software tool could be made for such similar projects which would be immensely helpful in the structuring and proper working of the project

<p><i>and Environmental Science.</i> Vol. 140. No. 1. IOP Publishing, 2018.</p>	<p>construction of University Sultan Zainal Abidin (UniSZA), Besut Campus, Student's hostel. CPM/ PERT analysis have the capability to be adopted in a scheduling system which involves the activities of the project, times and critical paths.</p>	
<p><u>Scheduling Project Management Using Crashing CPM Network to get Project completed on Time:</u>   <i>Mrs. Ruchita Shrimali Vyas</i>   <i>International Journal of Engineering Research &amp; Technology (IJERT) Vol. 2 Issue 2, February- 2013</i>  <i>ISSN: 2278-0181</i></p>	<p>This paper discusses about Crashing using a CPM Network in a project, which is basically bringing new resources to the critical path so that the project is executed in a more efficient manner. The planning and development of the Columbia Pkoana Precast Block Production Plant has been examined and a CPM workflow was used for the crashing of this particular motive. A few tasks such as Raw material testing and sampling , Loading aggregate by loader to four bean and so on was considered, for which CPM diagram was drawn, Critical path was found out, Time-Cost trade offs were calculated and minimal cost was calculated</p>	<p>A total cost of 840 mins was saved under the usage of CPM and this infers that usage of CPM can reduce time wasted and efficiently cut costs. A software that can calculate this would be greatly helpful.</p>

### 3. Overview of the Proposed System

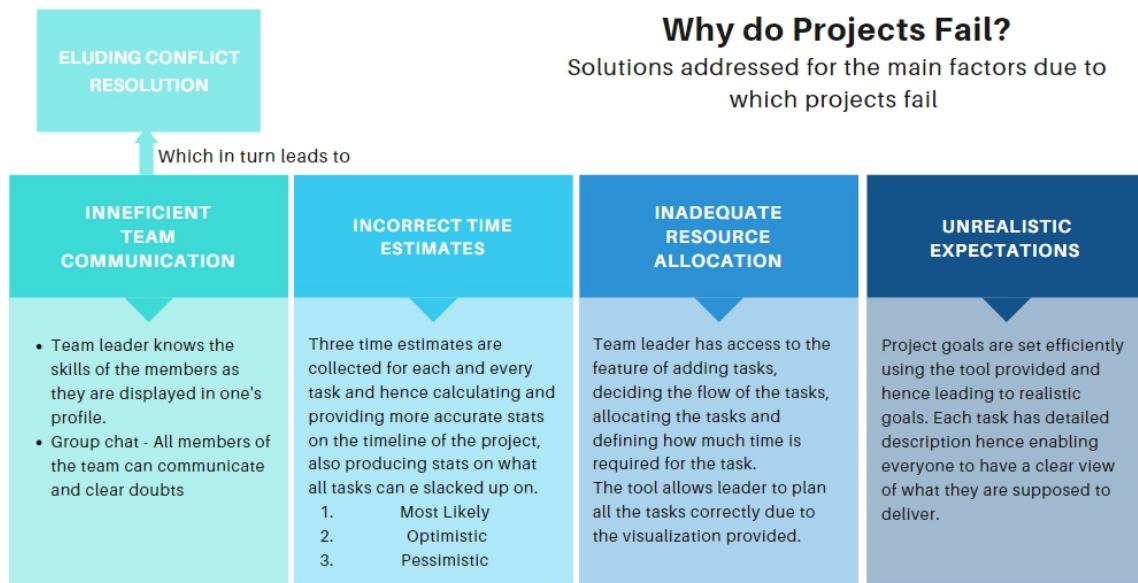
#### 3.1. Introduction and Related Concepts

This section will have a brief explanation about the concepts used to address each problem that was aimed to tackle and provide solutions to in this project management system.

The main reasons why a project fails have already been discussed earlier (refer to *Figure 2.0*). The solutions to tackle the problems listed before have been discussed here. It can be perceived that the following are the factors that affect the result of whether a project is successful or not:

- I. Communication
- II. Time Management
- III. Task Management
- IV. Resource Management (especially managing the people working)

The first problem which is inefficient team communication if not addressed leads to the second problem of eluding conflict resolution. To address this issue, a group chat feature has been developed. Here team members will be able to share their views, give updates and clear doubts with the team.



*Figure 3.1.1 Solutions to avoid project failures*

As for the other three problems, which are, inadequate resource allocation, incorrect time estimates and unrealistic expressions, this project attempted to address the issues with the well-established algorithms of PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method).

As mentioned above, PERT and CPM are used to make an estimated plan that can be comprehensive and volatile enough to adapt to changes. Projects are broken down into tasks and these tasks are interlinked in order to produce a series of connected links and thereby interdependent on each other. When tasks are interconnected, the times they are supposed to start and finish are also interconnected. The times at which they should optimally be completed are taken care of and monitored by PERT and CPM. When there are changes on the tasks, PERT and CPM are calculated dynamically and thereby an overdue is calculated, keeping the order and balance of the project in check.

### **Pert ( Project Evaluation and Review Technique ) :**

The Pert technique helps us by estimating the time taken for a project as a whole and thereby in a real-world scenario provide us a clear idea as when the project should be completed. If the schedules are followed the time used in the process would be done in the most optimal manner.

Considering each of the tasks that are present in a particular project, they are supposed to be presented with three different types of times, in terms of completion, being:

- Optimistic Time
- Most Likely Time
- Pessimistic Time

#### **Optimistic Time:**

This is the time taken in the consideration of a task having no drastic problems and thereby follows a smooth path in the processing. The most optimistic approach is considered here and the least time to finish a task is thereby the one considered.

### Most Likely Time:

This is the realistic time that is taken into consideration. Generally, any task would have its own problems to be faced upon which could be in the form of human error, lack of resources, lack of efficiency or so on. Hence a balanced take is put in the most likely time.

### Pessimistic Time:

A task that has a lot of challenges and if we disintegrate and check with those units in mind, a more melancholic time of the task can be produced. This time, which is more than both of the others, is what we consider to be the highest number of days the task can take.

### PERT Formula:

The pert Estimate for a particular task (E) can be calculated when we sum the Optimistic Time (O), 4 times of the Most Likely Time (M) and the Pessimistic Time (P) and divide the sum by 6. This is calculated each time there is a change in the Optimistic, Most likely and Pessimistic times of a particular task.

$$E = \frac{O + 4M + P}{6}$$

Estimated Time (E)  
Optimistic Time(O)  
Most Likely Time(M)  
Pessimistic Time(P)

Figure 3.1.2 Pert Formula

Additionally, if we require more statistics of the task, the standard deviation of a task can be calculated by Subtracting the Pessimistic Time by Optimistic Time and then dividing them by 6. This can be used to calculate a range in which the tasks can be allotted upon where in the estimated time + Standard deviation being the upper limit and the estimated time - Standard deviation being the lower limit of the range.

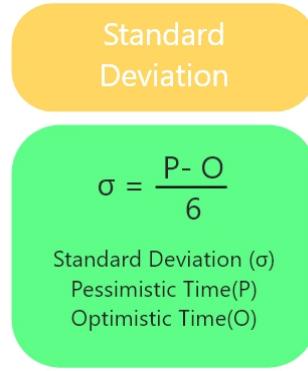


Figure 3.1.3 CPM Formula

### CPM (Critical Path Method):

This technique is used to calculate a routed graph for the tasks and to calculate the earliest and longest of the start times and end times for each one of them. Additionally it can be used to calculate the most important tasks that can alter the course of the project drastically called the Critical Path. Taking into account the critical path tasks and the other dependent tasks, a relative time is calculated for each task giving the amount of time that the task can be slacked upon.

These factors can be figuratively seen through an example mentioned below,

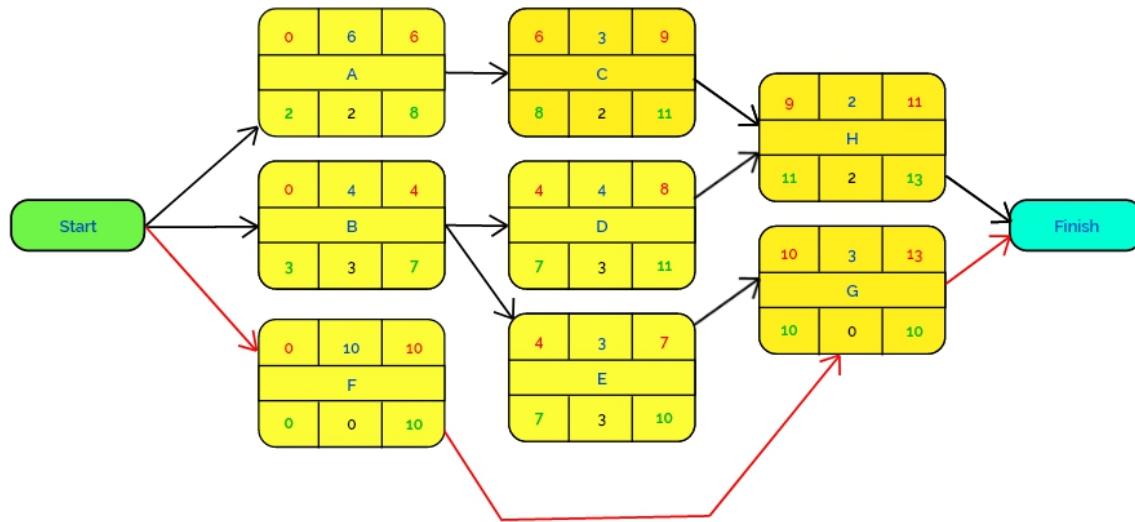
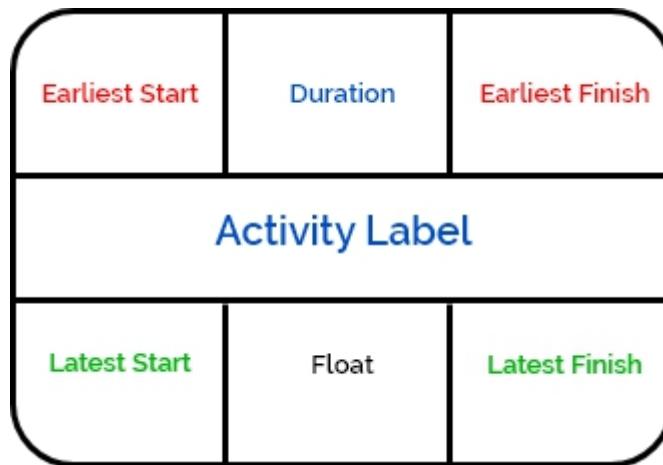


Figure 3.1.4 Pert Calculation example



*Figure 3.1.5 Mapping*

CPM calculations can mainly be split into 3 categories after the plotting of the graph:

- Earliest Start and Earliest Finish Times:

Looping through the tasks in a forward direction we calculate these times. This loop is called the Forward Pass.

- Earliest Start Time:

The predecessors with respect to the task in question are looked for their Earliest Finish times and the highest one is chosen. If there are no predecessors, the earliest start time for the task is taken as 0.

- Earliest Finish Time:

This can be calculated as the sum of Earliest Start Time and Time to calculate the current activity.

Project completion time can be taken by the highest of Earliest Finish and the Finishing node

- Latest Start and Latest Finish Times:

Making a backward loop from the last to first node, we calculate these times. This loop is called as the Backward pass.

- Latest Start Time:

Latest Finish Time of the task subtracted by the time taken to complete the task.

- Latest Finish Time:

The minimum of the Latest Start Times that come after the current node. If this is the last node, then the completion time is taken as the latest finish time.

- Assigning the Critical Path / Slack Times for the Tasks:

The slack times for each of the tasks can be calculated by Subtracting the Latest Finish time of the task and the Latest Start time of the task or by Subtracting the Earliest Finish time of the task and the Earliest Start time of the task. When this slack of a particular task becomes zero, it becomes a critical path task. The tracing from the start to the end of the tasks with slack zero is called as the critical path of the graph. When the slack becomes more than 0 it cannot be considered a critical path task, rather can be called a dependent task where in particular task/s have to be completed before these ones can be taken into consideration and thereby providing some buffer.

Project management tools using the above mentioned concepts have been built and incorporated in a website. The website has been developed using MERN Stack which is an amalgamation of the following four technologies: MongoDB, Express, React, Node.js. The front-end, the database, backend APIs and the various algorithms are going to work hand in hand enabling the website to function as a whole system. With this system, Javascript is used on the client side and Node.js on the server side. The main reason to choose MERN stack to be our basic System is because it provides the possibility of mastering both the algorithmic and logical part of the backend, as well as the design, user experience and animation components of the front end part. Also, MERN is one of the very popular systems used today to develop a good website as it consists of the very popular technologies which are very much required in the job market.

### 3.2 Framework, Architecture or Module for the Proposed System

The diagram below depicts the framework used to develop the whole system. The very popular library Mongoose takes care of transforming data from Javascript data into MongoDb and vice versa. Data manipulation becomes very easy along with efficiency and simplicity. Node.js works with JavaScript. These two languages are strongly linked and are able to interact with each other very easily. So, Node.js has been used to write the backend APIs. React.js is used to develop the frontend of the website and was chosen to be used as it is renowned for accelerating the loading speeds of websites in general.

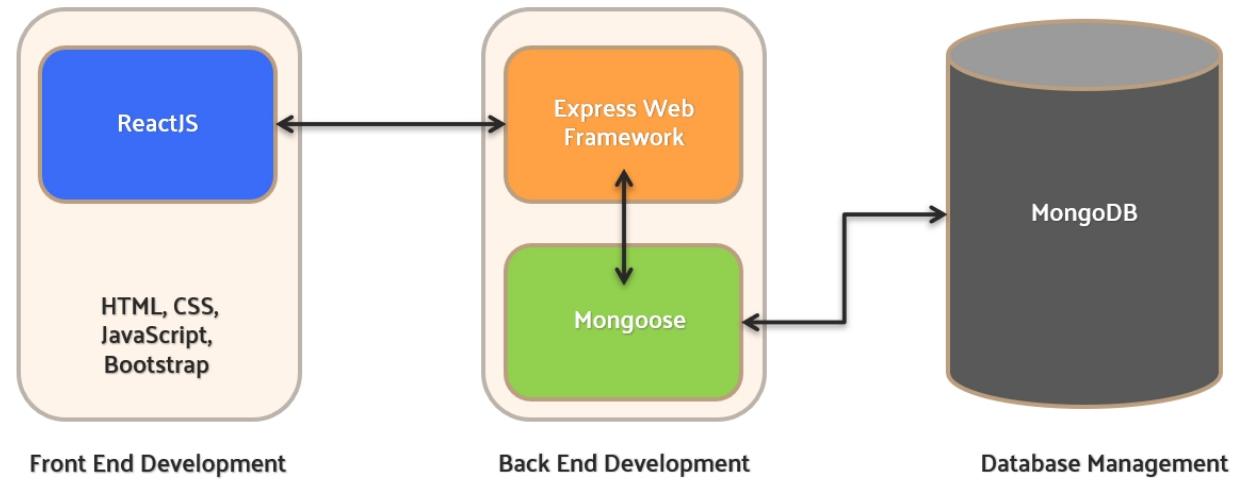


Figure 3.2.1 MERN Stack Framework Architecture

The project management tool developed can predict the following:

- I. Estimated date on which the project can be completed
- II. Critical path - the tasks that has to be worked with high priority and cannot go off schedule
- III. The tasks that can be worked in a little more relaxed manner - These tasks have slack time, i.e, these tasks have more days than originally allocated

## How does the project management tool help?

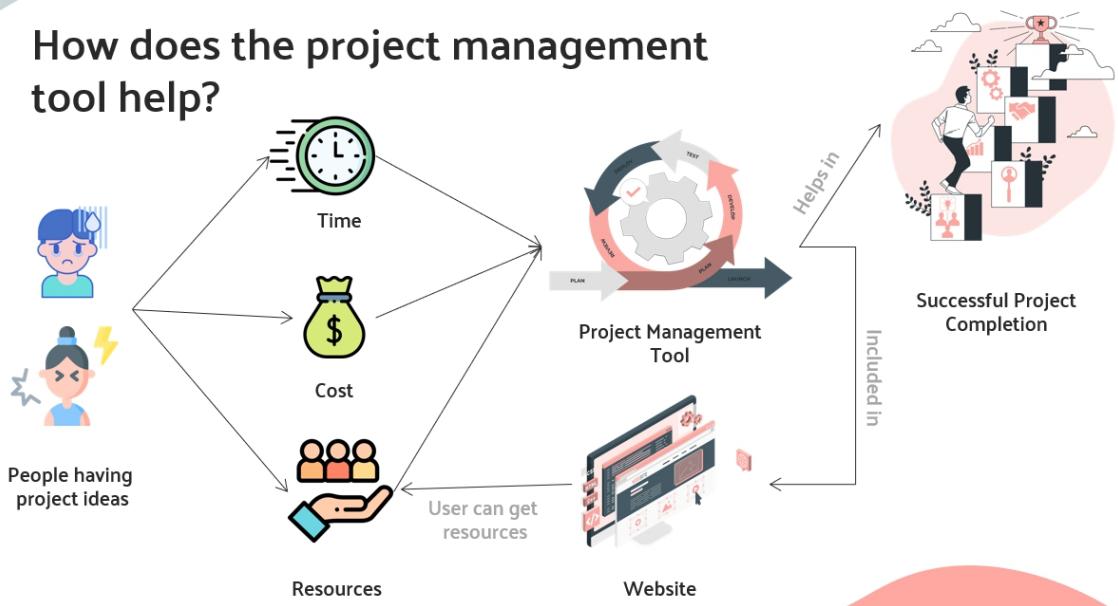


Figure 3.2.2 Project management tool integration

As discussed before, the main factor that leads to a successful project management is the efficient management of time, cost and resources. As the website developed acts as a networking portal too, individuals can find resources (in our case ‘people / users’) to join their project. So now, the individual has the time, cost and resources needed to start the project. But what about the efficient management aspect? That’s where the project management tool developed comes into picture. This tool takes in all the components of the project as the input, predicts the above mentioned and gives the statistics to all the team mates. Now that all members have a clear idea about the time periods, the task flow and their responsibilities, the project can go on with full momentum.

PERT and CPM are both project management techniques. Even though their functionalities do look similar as they both are network based project management techniques, which exhibit flow of sequence of activities and events, they both have many differences which are listed below:

*Table 3.2.1 : Difference between PERT and CPM*

<b>Comparison Category</b>	<b>PERT</b>	<b>CPM</b>
Definition	Technique used to manage uncertain activities of a project.	Statistical technique that manages well defined activities of a project.
What is it?	Plans and control time	Control cost and time
Estimates	Three time estimates	One time estimate
Management of	Unpredictable activities	Predictable activities
Focuses on	Minimizing the time required for completion of the project	Trade off between cost and time, with a major emphasis on cost-cutting
Activity Type	No demarcation	Demarcation of critical and non critical activities
Used for	Research and development projects	Construction projects (projects which have a clear view of the tasks (events) and the interdependencies)

After much analysis, it was decided that using PERT and CPM separately won't be providing all the benefits and features. Rather, it will be difficult for the users to decide between which method to use and not all users will know what PERT and CPM are. Hence, let us analyze what are the important factors, statistics and results needed for a student to initiate, monitor, track, work and complete a project successfully. There are two types of individuals this project management tool will have to support:

1. Individuals who have a clear visualization of all the aspects of the project, that is, they know what are the tasks needed, the flow of tasks, and which tasks are interdependent on each other.
2. Others who have a raw primary idea and who would want to figure out more about the direction in which the project will head to as the project progresses.

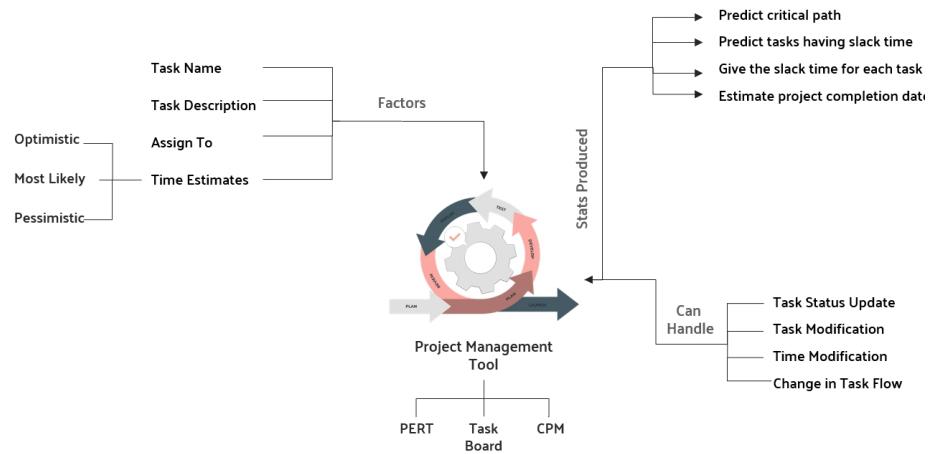
To develop a project management tool that will support both types of individuals stated above, the factors that have to be considered to develop this tool have been demystified and stated below. Basically project management tool must be able to accept changes and rework all the statistics to provide accurate and updated estimates and results:

- Add new tasks at any point of development
- Change / update the time estimates
- Edit task details
- Reassign task to a different member
- Reallocate tasks to a different member when one leaves

A project management tool solely developed with only either PERT or CPM will not be able to take in all these different factors, process them and produce useful results which are very much focused for projects managed and monitored by individuals (mainly students). Hence, to overcome this problem, only the features that were needed from both the algorithms were picked out and they are listed here:

- Tasks that have to be done with high priority. If these tasks are not completed by the due date, the project will not be completed on time. Basically predicting the critical path. - Part of CPM algorithm

- Predict the tasks that have extra time (tasks are not part of the critical path) - Part of CPM algorithm
- Earliest Start Time, Earliest Finish Time, Latest Start Time and Latest Finish Time of each task is calculated and set using CPM algorithm.
- Accept three time estimates from the user (optimistic, most likely and pessimistic) - Part of PERT algorithm
- Predict estimated date on which project will get completed - PERT algorithm



*Figure 3.2.3 Project Management Tool Functionality*

### Image Moderation:

When users tend to post some images either related to their project or not we need to be cautious on allowing it or not to show it on our website. We need to have a great amount of data and train it by ourselves from scratch if we need a very effective model to filter those images which are sensitive. Using a pretrained model is a good choice.

So we have used ResNet-50 which is a pre-trained Deep Learning model for image classification of the Convolutional Neural Network(CNN, or ConvNet), which is a class of deep neural networks, most commonly applied to analyzing visual imagery. It is 50 layers deep and is trained on a million images of 1000 categories from the ImageNet database. These models are then processed by tensorflowjs.

Content censorship can be implemented either on client-side (i.e., on an application itself) or server-side (i.e., on a remote server). In a client-side implementation, image filtering is done

inside of the application running on a user's device. In a server-side implementation filtering is done on a remote server.

In our system data would be processed before saving it to our database which is a remote server. That means the system should ingest data as it is being posted, perform processing and analysis to gain insights in near real-time. To achieve these capabilities, we designed an image filtering model which will be executed upon posting the images and will alert the user if any inappropriate content is encountered.

This filtering technique will try to analyse the image based on the model which gets processed by tensorflowjs. The main focus of this filtering is not to allow images which contain more than 90% of nudity or violent content. Once the image is loaded into our website, it would be processed through the checkNudity and checkViolence functions which processes the image with the models and produces an output of an object with keys nudity and violence which has a value for both.

If either of these values is greater than or equal to 90, it is considered inappropriate, else we assume the image isn't something with sensitive content. Based on the above value, the user will be notified as 'Inappropriate content' and 'Post created Successfully' respectively.

#### Image moderation:

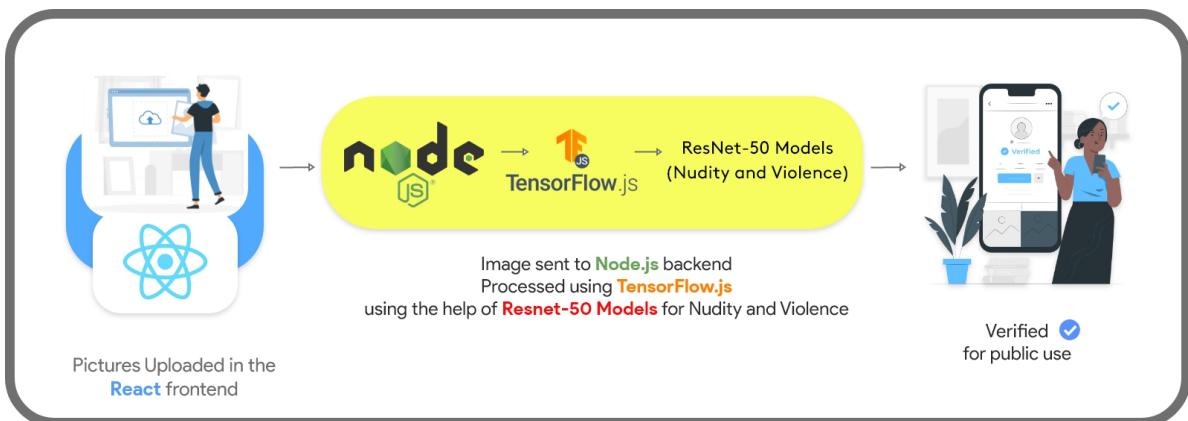


Figure 3.2.4 Workflow of image moderation

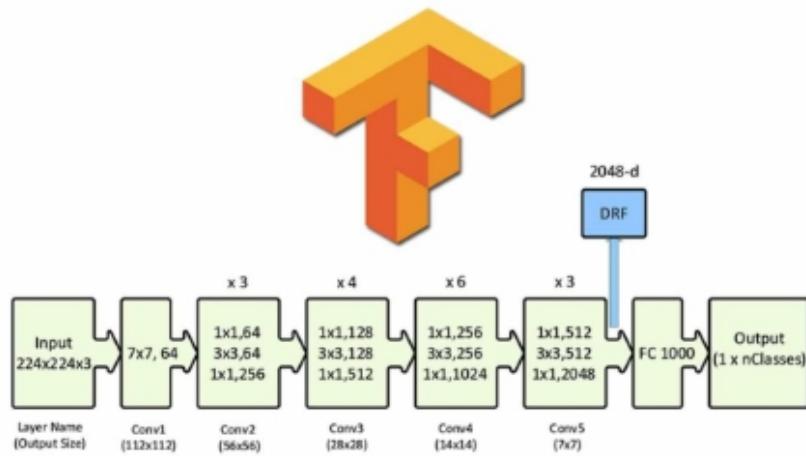


Figure 3.2.5 Tensorflow and resnet-50

### Video Moderation:

When users of the website tend to post videos, they may or may not have a malicious intent in mind. These videos might tend to hurt or offend the others who would like to use the social network for their progress potentially. This fact cannot be ignored and the fact that this social space must be used only for a person's well-being , organization and a healthy connection building, moderation is bound to take place. This moderation is done before the upload of the video into cloudinary for the post.

The video moderation is carried out in a simple process consisting of two steps:

- Taking samples from the video that is processed to figure out whether it is suitable or not
- The actual processing of the samples that can produce the percentage of Inappropriateness of the video.

How does this take place?

First the video is split into 5 images at 20%, 40%, 60%, 80% and 99% of the video and stored in a folder. Once this is done, the images are further processed using the method mentioned above for images.

The images are processed through the checkNudity and checkViolence functions which processes the images with the models and produces an output of an object with keys nudity and violence which has a value for both.

Considering this value, if it is greater than or equal to 90, it is considered inappropriate and is returned with the object and a status of 400.

If it is less than 90, we assume the image isn't something with malicious intent and thereby a status of 200 with a go sign is returned.

What process is used to check using the Model?

2 models, resnet-50 based (for violence and nudity respectively) suitable for processing by tensorflow are taken and these models initially in .h5 format are converted to .json format for use by tensorflowjs along with shards.

These models are then processed by tensorflowjs (node version) and loaded into variables individually in different functions. The videos, split into images are passed into these functions through their file paths, converted to Buffers (using Sharp package) and finally processed.

The model accepts an input of [0,224,224,3] where the input tensor stands with height and width of 224 and a batch size of 0 (Considered only one image process). Once fit as proper inputs gets called into the model.predict() and the obtained tensor with a [1,1] output, is stored in an Array and the first value is the output of the model being the percentage of nudity/violence normalized (0 to 1 Value).

This is returned and accessed in a function that creates the object with the percentage of violence and the percentage of nudity and this is used to check the image returned as an array along with the processing results of the rest of the images.

## Video Moderation:

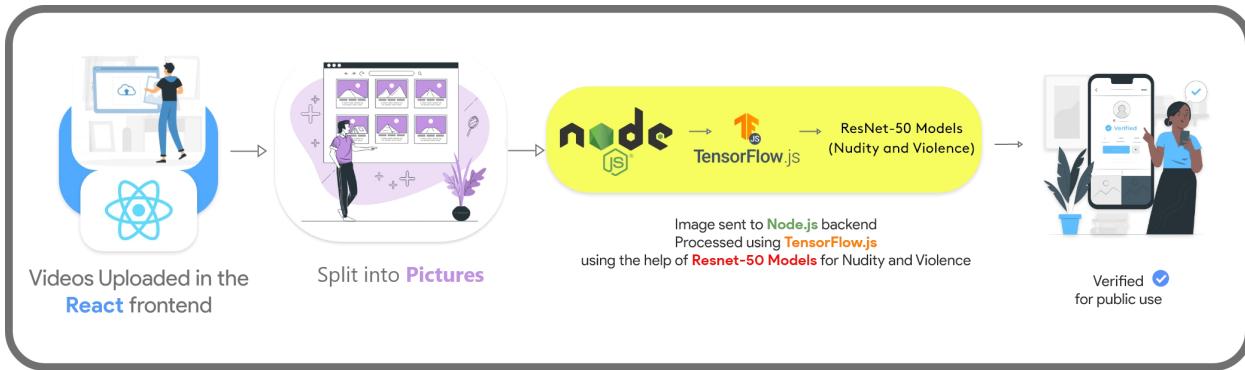


Figure 3.2.6 Workflow of Video Moderation

## 3.3. Proposed System Model

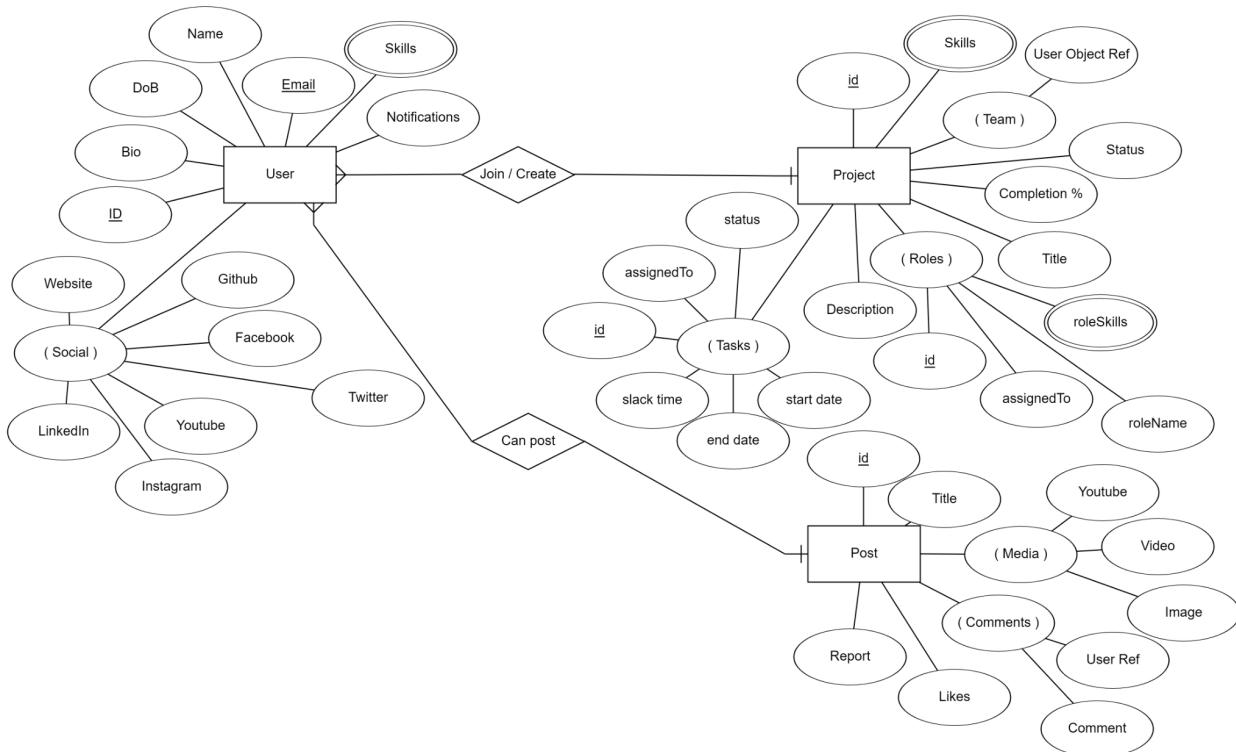
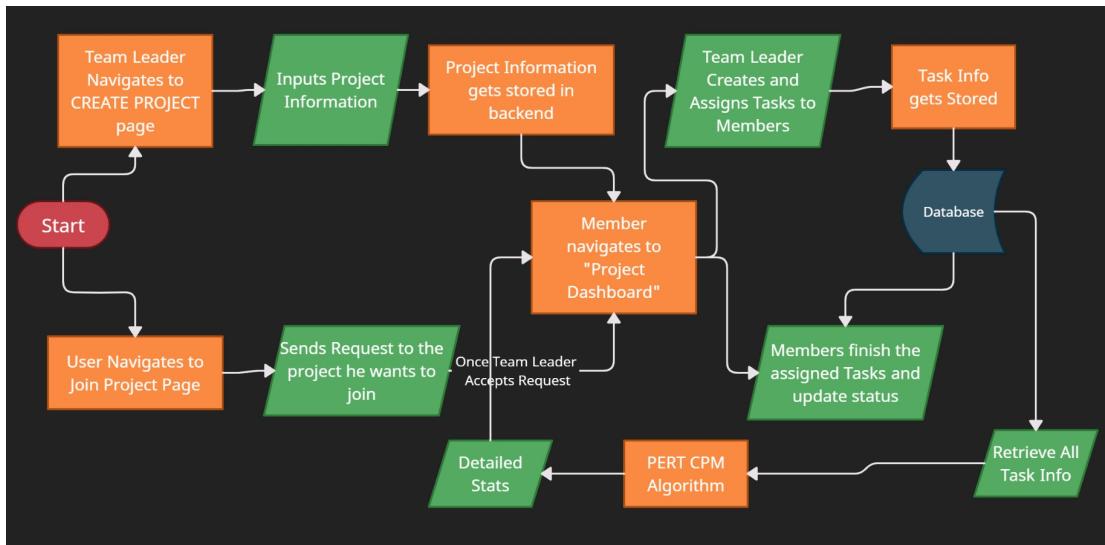
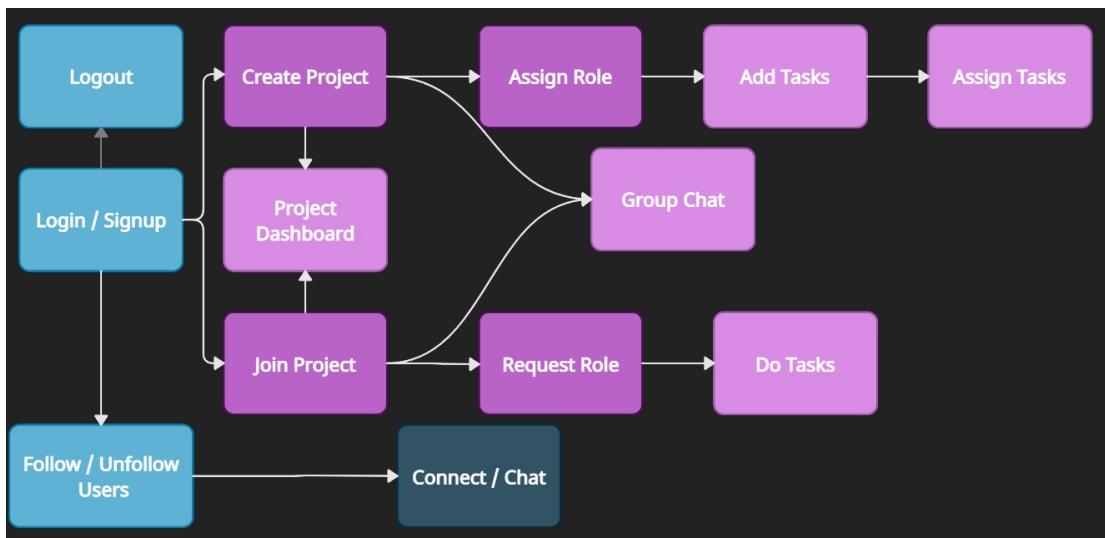


Figure 3.3.1 Entity Relationship Diagram



*Figure 3.3.2 PERT CPM Component Integration*



*Figure 3.3.3 All Modules Connected Overview*

## **4. Proposed System Analysis and Design**

### 4.1 Introduction

An idea in one's mind requires two things, a proper plan and a proper team to become a merit, be it a project, a start-up, a research paper or generally said to be a manifestation of a thought.

This system is designed in a manner so that we can implement a Website that people can access in a simple manner. This website has all of these free tools that can very well be used in the process of creation, management and publishing of a project. These tools used in an efficient manner can produce an efficient project, conserve time, can build relationships with like-minded people, polish one's team management/playing skills, grow technical knowledge and be advantageous in many other manners.

The system aims to provide a platform for tools such as a Project Dashboard, Task Manager, Group chat, Critical Path Calculator, Pert Estimator, Overdue Checker, Posts for each project and Notifications for each Member on the updates of the project. These said, being the project management tools, It also allows a user to follow other users, contact them through messages, post about their projects / personal achievements, and have profiles to maintain their digital footprint.

In the implementation of a Website, this can be used under any scenario as it can be accessed from any place, any time, since we can access it through any device under the catch that the device is connected to the Internet, which is easily attainable in today's world by a layman.

The system has been analyzed in a much detailed format below.

### 4.2 Requirement Analysis

#### 4.2.1. Functional Requirements

##### *4.2.1.1. Product Perspective*

The proposed Project Management System will be an online website. This system will allow students from different colleges to collaborate and work on a project. Through this system, we provide the estimated completion time of a project with slack time for each task defined for a project by the team leader. We will store the

following information in our database which will be given by the users (both project managers and the customers):

- User Details
  - Personal Details:
    - Name
    - Date of Birth
    - Bio
    - Location
    - Skills (will be captured in the form of tags)
  - Social Details:
    - Personal Website
    - Github
    - Youtube
    - Facebook
    - Twitter
    - LinkedIn
    - Instagram
  - Login Details
    - Email
    - Username
    - Password
- Project Details
  - Title of the project
  - Description of the project
  - Skills required for the project
  - Adding roles / positions
    - Role title
    - Skills required for the role

#### *4.2.1.2. Product features*

The major feature of this system are:

- Time and task management can be done with ease due the project management tool developed using PERT and CPM algorithms. The tool helps the user track the following:
  - The estimated date of the project completion
  - The tasks that have to be done with high priority and no lag as they are critical for the on-time project completion
  - Tasks that can have slack time and done at a slower pace
  - A network diagram that shows the flow of tasks
  - Give a clear view of the critical path
- Group chat feature for all the members of the group to stay connected
- Team leader manages who he/she wants to assign for the particular role of the project
- The team leader creates the workflow and assigns the tasks to the users
- Find people who have the skill set you need for the project, connect and collaborate with them

#### *4.2.1.3. User characteristics*

There are two types of users on this website and they have different user privileges. The functionalities and features accessible to either of the users are listed below:

- Project Manager / Team Leader
  - Project Creation
  - Project Deletion
  - Addition, Deletion and Modification of process (task)
  - Assign tasks to team members
  - Have control of the Dependency Graph / Define the flow of tasks
  - Review and mark tasks as Completed
  - Provide ratings to team members once the project is completed
  - Invite the most suitable person for the role

- Kick out the team member if he/she doesn't possess the knowledge required for the role
  - Group chat
- Team Member
  - Join a Project
  - Update tasks assigned to him/her
  - Leave Project
  - Group chat

#### *4.2.1.4. Assumption & Dependencies*

The product needs the following third-party products.

- MongoDB Atlas which acts as the database for our system
- Node.js needs to be installed for the back-end purpose
- React needs to be installed for the front-end purpose
- Cloudinary access which is needed for storage purposes(images and videos)

#### *4.2.1.5. Domain Requirements*

Our system mainly focuses on calculating project completion time. The Pert technique is one that helps us by estimating the time taken for a project as a whole and thereby in a real world scenario provide us a clear idea as to when the project should be completed. If the schedules are followed the time used in the process would be done in the most optimal manner.

Considering each of the tasks that are present in a particular project, they are supposed to be presented with three different types of times, in terms of completion, being:

- Optimistic Time - This is the time taken in the consideration of a task having no drastic problems and thereby follows a smooth path in the

processing. The most optimistic approach is considered here and the least time to finish a task is thereby the one considered.

- Most Likely Time - This is the realistic time that is taken into consideration. Generally, any task would have its own problems to be faced upon which could be in the form of human error, lack of resources, lack of efficiency or so on. Hence a balanced take is put in the most likely time.
- Pessimistic Time - A task that has a lot of challenges and if we disintegrate and check with those units in mind, a more melancholic time of the task can be produced. This time, which is more than both of the others, is what we consider to be the highest number of days the task can take.

And average expected time is calculated by taking a weighted average of these three points of estimates using the formula below:

$$E = \frac{O + 4ML + P}{6}$$

Where,

- E is the Mean PERT Average,
- ML → *Most Likely Time*
- O → *Optimistic Time*
- P → *Pessimistic Time*

$$SD = \frac{P - O}{6}$$

Where,

- SD is the Standard Deviation
- P is Pessimistic Time
- O is Optimistic Time

#### *4.2.1.6 User Requirements*

Our portal provides a platform through which users are able to interact with students from different colleges. Once the user has a creative idea he/she can start his own project which must be unique from the existing project and invite students from his/her own or various colleges. For this, we have developed a model which recommends the most suitable profile for the roles he/she defined in his/her project. Once the invitation has been sent, the concerned person can either accept or decline the request. And a student who is willing to join that project can send a request and the team leader can either accept or decline by viewing the student's profile. Once the project group has been formed the team leader can create and assign tasks to all the team members based on skills he/she is expertise in. And the workflow will be created by the team leader followed by which pert calculation will be done internally and the system will set the deadline for that project and intimate users about the tasks which have the highest priority and must be completed in time so that it will not cross the expected deadline. And all the team members and leaders can interact with each other through the group chat portal which will be enabled upon the creation of the project to stay connected. And they can also interact through individual chat if anything matters personal. Once the tasks have been assigned to each and every member they will work on them and they will be provided with a deadline of some days by the team leader. If the task procrastinates the leader will get notified about that and an alert will be shown on that task card. If the task gets completed the member assigned to that task can switch his task card from progress to review. Then the leader will review his/her work and mark it as completed. If the work is not satisfying the leader can move the task card back to progress and intimate the user about the changes to be

made through the chat facilitated. If all the tasks are marked as completed the leader can finalize the project and mark the project as completed. If the project gets submitted no tasks can be added, deleted, or updated. Once the project has been completed the leader can provide feedback through ratings(1-7) for all the team members which will be considered while recommending users to other projects.

If the team leader finds the member in his/her team is no more suitable for the position he/she is assigned, the team leader can kick out that user and assign someone who is more capable. And if the team member finds it difficult to work on his/her project they can leave the project and the task assigned to that person will be assigned to the team leader temporarily. Once the new person signed in to the project they will be assigned to that position.

Our system is designed in a way that allows each and every person to get involved in at most 5 projects so that the person will engage in all of his/her projects. Apart from this, the users can post some images, short demonstration video, text, youtube video related to his/her project or something which he finds valuable to share. Others are able to like, comment on the post he/she sees in their feed. If any of the users find the post good, he/she can share it with their friends/relatives and they can like or comment on the post once they sign up to our website.

#### 4.2.2 Non Functional Requirements

##### *4.2.2.1. Product Requirements*

*4.2.2.2.1. Efficiency (in terms of Time and Space)* - The website uses the React.js framework for the frontend. Accompanied with the Redux this provides an extensively less strained way of producing the visual components of the system. The state management provided by the Redux and React.js combo, wherein the Independent components can manage with states from React.js and dependent ones can easily access the states required for the UI from the global storage provided and updated by Redux, turns out to be a boon in the loading times of the

website and there providing multiple places of reduction in lag and an interactive interface for the end-user. The space consumed by these files (excluding the node modules which will mainly be stored on the hosting site after installation on the server) costs very little and equates to simple javascript files that we generally create, each file costing Kbs of space.

The usage of Socket.io for the message also allows users to interact seamlessly with no lag in time if the server is accessed by the clients in a proper way. Client-to-client communication is provided by the means of the socket.io package and socket id's. This system allows instantaneous action of the messages sent by a person and received by another.

The strain presented by the whole system might be a little lot to handle under a local system with simple functionalities. But since we consider hosting the API and the frontend in Heroku / other websites that offer web hosting, the strain is laxer and the servers take up the load. Thereby, the loading times of the website is efficiently loaded and these can be seen effectively when the project is in live state.

**4.2.2.1.2. Reliability** - The whole project was maintained in a proper way on GitHub using Version control and commits. These measures made a very effective monitor on the bugs and sharing of code. The bugs were then and there looked onto and the safe maintenance was preserved of the code in order to provide as smooth as a User experience possible.

Since the user interface is based off React.js and Redux, it is much more reliable than normal HTML, CSS and Javascript as we get a chance to log the errors and manage them more easily in terms of Logging of them.

Node.js was used for the backend part and the nodemon package, monitored and kept displaying errors, thereby a live outlook of the errors made it much easier to debug and this kept the project in check for a proper experience when resorting to the requests made from the frontend.

Overall a very smooth experience and reliable system is made under high monitoring and even if there occurs an error in the database, since it is on the cloud (MongoDB Atlas) it can be easily logged into and managed by one of the admins.

**4.2.2.1.3. Portability** - Since we considered the fact that a Lay man must be able to access the application just the way someone who lays on a better financial background, we considered the application to be a Website as it loads and works much easier than any other alternative. Contrary to popular belief that websites are usually much more difficult to use than an application in a phone, when the UI and UX is taken care of it can turn out to be an equal competitor to an application. Thereby we focused on the layout of the website which can be accessed through any kind of device. Bootstrap used in the website helps it to be resized properly for any device such as a Phone, Tablet or Laptop/Desktop. This approach thereby helped in the portability of the application.

In today's fast moving world most people have access to a smart device, and due to the efficient plans offered by internet service providers, accessibility to the Internet. Considering the fact that 5G will take over the world faster than we can imagine, the availability of networks will definitely be more attainable in the upcoming years. Hence, we can definitely access the website from a Lay man's perspective.

**4.2.2.1.4. Usability** - The website as a whole follows the current day social media platform trends for its User Interface and the Project Management tools are very easily understandable and accessible in the UI. People will not have any difficulty in using the website or learning about it as it tends to follow a simple yet efficient approach in user experience.

Since the tech stack considers the use of Redux in the state management and loading, the UI provided makes a Heavy impression using the speed. The requests that are sent to the backend are also carefully looked upon so as to not unnecessarily produce a lag in the loading and never to produce a probable

infinite loop that may cause the website to overflow in terms of space and processing power.

Considering the fact that flaws may still appear in the terms of learning how to use the website, a tutorial will be made soon and available through a playlist on youtube so as to cover the nooks and corners of the website.

#### *4.2.2.2. Organizational Requirements*

**4.2.2.2.1. Implementation Requirements** - The website so run on Webpack plugin for React can be straight hosted into a development site that hosts and supports such a Webpack Package. This is considered to be straight solved by heroku in an efficient manner and a cost free one too.

The website uses a MongoDB Atlas for the maintenance of User records, Project records and Post records. This MongoDB Database and its collections can be managed by the Admin by a simple login. Updation, Deletion and Addition of flawed, duplicate or unnecessary records can be managed by the same.

To store the images that the Users post and the Videos such posted, A Cloudinary account is separately maintained for the both and optimized images along with videos are allowed to be added and posted in these. The urls obtained from the responses of such api creation calls are used to be stored in the mongodb database and these URLs are accessed by a request from the frontend to backend when in need. A Cloudinary free subscription allows us to use 25GB of space and if more is needed, a small fee is required.

Apart from these, the API's that were used for the Functioning of the website cost nothing and are free to access. However, some API's such as Youtube API demand a fee after the limit that is free (10000 calls allowed per week).

Hence, a truly cost free approach is now used in the making of the website, but in terms of a real time scenario when we get to host a website for Live users, costs might be considered in a marginal way.

**4.2.2.2. Engineering Standard Requirements** - Under the consideration of future scope for the project, A Trademark can be obtained under a particular name and thereby, the license to use the name for any type of an application can be done. The web app considered now can be brought into multiple sectors and an application can be done for the same.

The Website can be brought into contemporary sense in the daily life of a student who wishes to achieve something by bringing his/her ideas to life if we can produce this in the universities under which he/she studies. This can be a real game changer for the way the projects are presented and thereby a timeline of the project can even be presented in business pitches under the consideration of entrepreneurship with the idea. Also, research papers can be built with the website and it can also be brought into limelight with the website and the project's timeline.

For the credibility of the website, the Website's development git page can be looked into. (This is currently private and can be accessed only by the admins who govern the website)

#### *4.2.2.3. Operational Requirements*

- Economic - Since the website is made and maintained by us, there is no development costs as such. The costs of Heroku (in case there is a necessity of space to run frontend or backend) or any other Website Hosting platform, Database (in case there is a need for space in the Atlas MongoDB database, chances are very less) and the costs of the Cloudinary accounts ( the Accounts that are used to store the Images and Videos, 25GB per month is allowed to be accessed by the users as a whole for images and videos separately).

In case there is a need to produce the website as a Startup there will be costs for marketing which is suitable enough to reach the target audience involved.

- Environmental - On a Micro level, from the end user's perspective only a certain simple need is present in order to access and use the website. A Stable internet connection and a device such as a Laptop/Desktop or Mobile would be more than sufficient to access it.

Since we take care of the server side work on heroku, we do not need particular hardware to host the website for itself. Rather, we would need to think about the heroku resources and decide accordingly for the website.

- Social - The project mainly involves a target audience of 1st to 4th year students in colleges who would like to access and work in the platform. Apart from them, we can also have teachers, PhD scholars, M.Tech students, Free Lancers, People in jobs or just a person who wants to make his idea a real deal.

Considering this fact, the users can mainly be split into 2 categories:

- People who Host projects - The people with their ideas can post their own projects and list it down to the users who would like to collaborate with them. These people would be considered the leaders of the project and they would take full charge of recruiting, managing and kicking out people. They will also have to keep a tab on the tasks and the flow of the project so that there is a smooth processing and fulfilment of the same.
- People who want to collaborate - Considering people who want to use their time productively, the website serves to provide a list of ideas that they can collaborate with other people and work on. These people can monitor their own tasks and their workflow. They can ease the load on the leader as they can make changes after communicating with the leader and thereby teamwork is preserved.
- Political - The idea was implemented as a website just so that it is accessible by any person at any point of time, anywhere in the world.

Thereby hosting the website on a hosting site, will make it very tangible to access from any remote location.

- Ethical - The website has a filtering system that serves to have an eye over what a user is posting and reject unwanted material that can offend other people. A sentiment analysis is done over the text posted over each post and pictures and videos which are posted are looked upon first before the upload to check if it is allowed to be uploaded for public view. Also, users can report unethical acts that are shown through posts by a report system in the website. The reporting of a post with more than 3 users can lead to the removal of a post, which will be notified to the user in charge of posting it.
- Health and Safety - No particular ways of affecting one's physical health can be seen in the website. The usual terms that are followed in terms of a person viewing a website must be followed in order to preserve themselves from any strain and effect on their eyes.

Mental health is carefully monitored, since negative content is generally not allowed to be posted on the website and thereby chances of getting affected mentally is very less.

- Sustainability - The users who use this website will have a sense of urgency to report to the project members or at least update their own tasks as it provides a sense over when the project will be completed upon.

The posts and regular updates for the social community through Notifications for Projects, People and other happenings can cause the necessary focus towards leaning on this platform making it sustainable under efficient end user use.

- Legality - The Website if looked under future reference, a legal perspective requires the website to be trademarked under a particular name. This will also allow the team to publish it and market the same legally. Additionally, we can also make an application for the ease of a mobile

user under the same name. The trademark serves to be the legal uphold and license to promote the website. It also saves us from others who might try to use the name.

- Inspectability -

- The database of the website can be made available to anyone who wishes to access it, with prior permission from the admin team with the user credentials on the mongoDB Atlas website.
- The Cloudinary account is also monitored under an account which can be logged into by the admin with the credentials to access the same. A separate account is maintained for the Images and a separate one for the Videos.
- A Github page is privately available to the admin of the website and this can be accessed by the users with prior permission from the admin team.

#### 4.2.3. System Requirements

##### *4.2.3.1. H/W Requirements(details about Application Specific Hardware)*

- ❖ Operating system: Windows 9x/XP,Windows 7/8/10
- ❖ Processor: Pentium 3.0 Ghz or higher
- ❖ RAM: 512Mb or more
- ❖ Internet : Stable connection

##### *4.2.3.2. S/W Requirements(details about Application Specific Software)*

Client Side:

- ❖ Browser: Edge, Chrome, Firefox or Safari

Admin Side:

- ❖ Node.js proper version installed and running properly
- ❖ Packages installed using npm
- ❖ React application
- ❖ MongoDB Compass to check Database

## 5. Results and Discussion

### 5.1. Project Dashboard

A project taken into consideration has a lot of factors to deal with which includes the team information, statistics of the project, task management, network diagram for tasks, group chats and posts. These are collaborated and brought together in a single panel called as project dashboard.

Initially a Project's title and description are mentioned in the top as shown below, (This project has title newone and description newone):

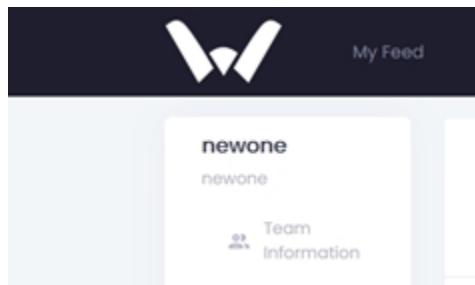


Figure 5.1.1 : Project Dashboard

After which comes the components that need to be taken care of. These components are listed below and explained:

#### 5.1.1 Team information:

The project details that include Project Name, Project Leader's name, Start date, End date, Progress of the project and the Roles of the project (if filled up by a person, his details or availability to choose people from the most suitable backgrounds through a modal popup is shown).

The screenshot shows a project management interface. On the left, a sidebar for 'newone' displays options like 'Team Information', 'Project Stats', 'Add Task', 'Network Diagram', 'All Tasks', 'Group Chat', and 'Posts'. The main area is titled 'Team Information' and shows a list of team members: 'newone' (Akash), 'newone' (Akash), and 'newone' (Akash). It details a task for 'NET Remoting' with a start date of 31-05-2021 and a due date of 03-06-2021. A progress bar indicates 0%. Below this, a table lists roles: 'z' (NET Compact Framework, assigned to Akash), 'testrole' (HTML, assigned to Akash), and 'testrole2' (HTML, assigned to Akash). A button 'Send invite' is shown next to each role entry.

Figure 5.1.1.1 : Project - Team Information

### 5.1.2. Project Statistics:

This includes project statistics, such as Start Date, End Date (Estimated from pert) , Days left, Number of tasks in the Todo Section, Ongoing Section, Review Section and the Completed Section. Along with these details this component includes a Critical Path Task section that showcases the tasks that are most crucial in the determination of the project completion along with their details being status, due date and the person they are assigned to. If there are any tasks that can be slacked upon, those slacks are considered and their slack times are mentioned too.

The screenshot shows a project management interface. On the left, a sidebar for 'hello' displays options like 'Team Information', 'Project Stats' (which is selected and highlighted in blue), 'Add Task', 'Network Diagram', 'All Tasks', and 'Group Chat'. The main area is titled 'Project Stats' and shows analysis of tasks. It displays the following data: Start Date: 30-05-2021, Days Left: 11, End Date: 15-06-2021. Below this, it shows task counts: To Do Tasks: 1, Ongoing Tasks: 1, Reviewing Tasks: 2, and Completed Tasks: 1.

Figure 5.1.2.1 : Project Statistics

### 5.1.3. Add task:

Every task requires a title, description, people whom it is assigned to, optimistic, pessimistic and most likely time, so that calculations on pert are done smoothly. These are added in this separate component called “Add Task”

The screenshot shows a 'Create Task' form within a web-based project management application. The left sidebar has a dark theme with white text and icons. It includes links for 'My Feed', 'Users', 'My Projects', 'Create Project', 'Join Project', 'Chats', 'Notifications', 'Recommendation', and a 'Hi, akash' button. The main content area has a light gray background. At the top, it says 'Create Task' and 'Add Tasks and allocate to members.' Below this are fields for 'Task Title' (a text input), 'Task Description' (a text input), and 'Assign to' (a text input). At the bottom, there are three input fields for 'Optimistic Time', 'Most Likely Time', and 'Pessimistic Time', each with a small calendar icon. A blue 'Create Task' button is at the bottom right.

Figure 5.1.3.1 : Creation of new tasks

### 5.1.4. Network Diagram:

This diagram is made to connect the tasks that we currently have and the tasks in the project can be connected sequentially or disconnected here. This also gives a chance to view the planning of the project visually and more clearly.

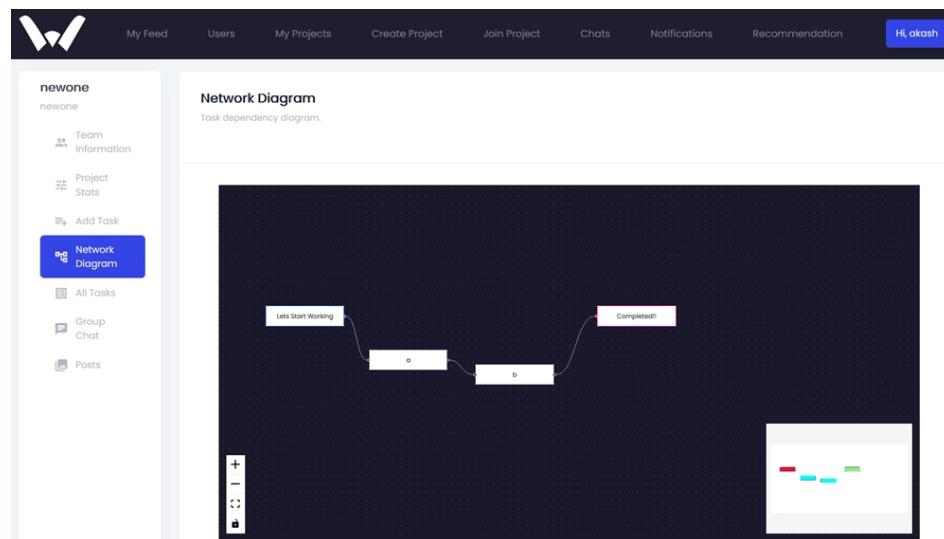


Figure 5.1.4.1 : Project Network Diagram

### 5.1.5. All Tasks:

This component is used to show the tasks in order of progression. All the tasks are classified into To Do, Work in Progress, for Review and Completed.

The Users who are assigned to these tasks can move the tasks freely between To Do, In progress and For Review. The Leader must confirm whether the task is properly completed and move it to Completed Status

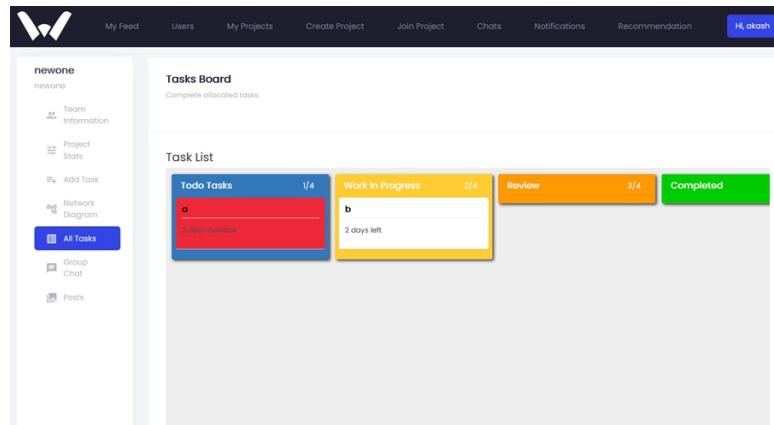


Figure 5.1.5.1 : Project Tasks Board

### 5.1.6. Group Chat:

This is used for the communication of the group members. Real time communication is enabled through Socket.io and a very less lag can be experienced by the team.

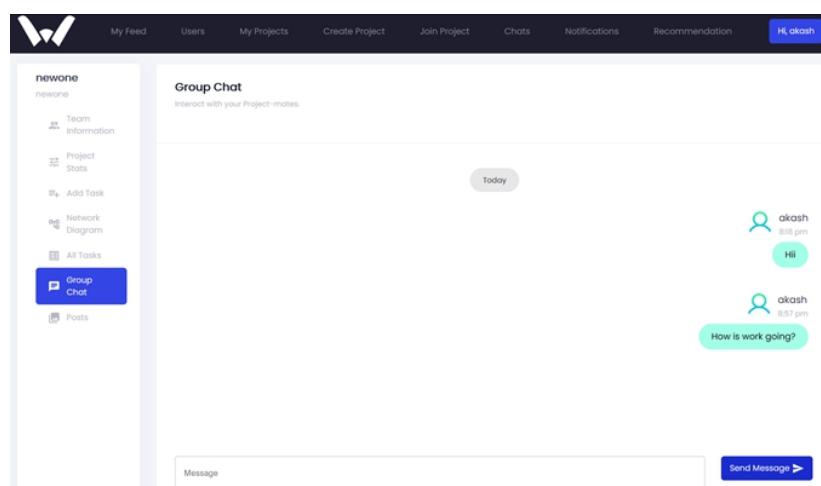
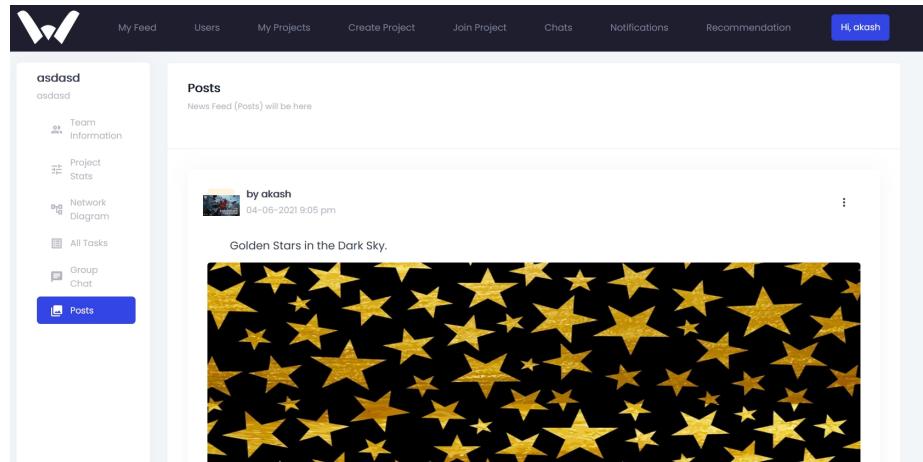


Figure 5.1.6.1 : Project Group Chat

### 5.1.7. Posts:

Posts that are correlated with the project in the time of posting, are shown with respect to the projects here.



*Figure 5.1.7.1: Posts tagged by project*

Thereby a project is fully and exclusively looked upon and its statistics along with the other details are shown in a manner that a person can access and edit to his needs.

## 5.2. Dealing with Projects

In our portal a project can be Made, Edited and Managed by a user. The person with an idea can Create a project that can be viewed by other users and Requested to be accepted upon by this user. The people who want to join the project and work on it once accepted have this project listed in their my project section ( A Leader gets this once he creates the project )

These pages are explained after categorizing below:

### 5.2.1. Create Project:

Here one can create a project using his ideas, this can be accomplished when one clearly states his title, description, skills involved, roles, role skills respectively for the project. This can be seen through an example below:

Let's Start a New Project  
Fill in the form with all the necessary details to register the project.

---

Title of your Project  
Workshake

Description of the Project  
Portal to Connect, Learn, Manage, Achieve and Grow :)

Skills  
React.js Node.js MongoDB Redux Javascript HTML CSS Npm

Role Title Frontend Developer	Role Skills React.js HTML CSS Redux	<b>+</b>
Role Title Backend Developer	Role Skills Node.js Javascript Npm MongoDB	<b>-</b>

**Create Project!**

Figure 5.2.1.1 : Creation of Project

Once this is done, the project can be listed as one which is live and people can look into details of the project and request for it

### 5.2.2. Join Project:

This is the page where a person who wishes to look into projects and join one can search for the most suitable projects for him and pick one from the list. Once satisfied he can request for a role in the project and if this is accepted he is part of the team and will be assigned tasks by the leader.

Join Project  **04 June, Friday | 10:10 pm**

Role Name	Skills Required	Status
a	.NET CLR	<b>Request</b>
b	.NET Remoting	<b>Position Full</b>

Role Name	Skills Required	Start Date	Due Date	Progress
asdasd	okash	30-APR-2021	02-MAY-2021	<div style="width: 100%;">100%</div>
dsadadsa	okash	30-APR-2021	02-MAY-2021	<div style="width: 100%;">100%</div>

Figure 5.2.2.1 : Joining a project

### 5.2.3. My Projects:

Projects that contain our name in the team are listed in the My Projects section, where in the person could either be a member or a leader. These projects can have their dashboards accessed here, Edited, or deleted / left (member can leave, leader deletes) or submitted. These are categorized into Ongoing, Overdue or Completed projects and respectively assigned in the display. Invites, Assigning to roles, Kicking out can also be done in this section.

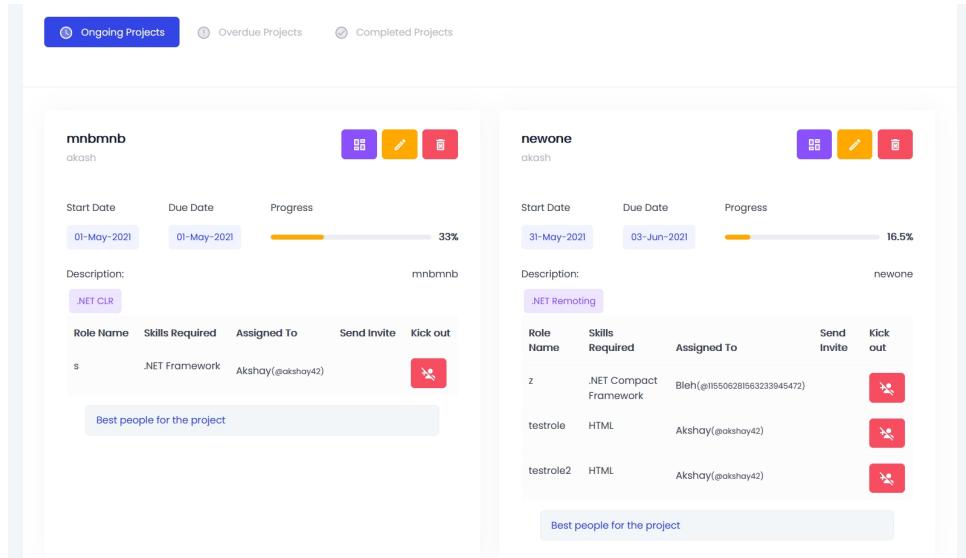


Figure 5.2.3.1: All Projects of a user

### 5.3. The Networking Platform

Along with the project management tool and project features discussed above, this website also provides a networking platform. By keeping one's user profiles updated, users can get more project invites and meet like minded people.

The screenshot shows a user profile for 'Nithiya K' (@WarriorNitz). On the left, there's a sidebar with a photo placeholder, 'Edit Profile' and 'Delete Profile' buttons, and sections for Email (nithiya070399@gmail.com), Location (Singapore), and Joined (Tue May 25 2021). It also lists 2 Following and 3 Followers, and links to Personal Information, Social Information, Project Stats, and Posts. On the right, a 'Personal Information' section titled 'Update your personal information.' displays the user's details: Name (Nithiya K), Email (nithiya070399@gmail.com), Bio (One of the Creators of Workshake), Skills (HTML, CSS, React.js, Node.js, Javascript, Java, Python), and Date of Birth (Sun Mar 07 1999).

Figure 5.3.1 : User Profile - Personal Info

Users can keep their account updated through “Edit Profile”. All the information will be updated to the database with the help of the APIs written.

The 'Edit Profile' form allows users to update their account details. It includes fields for Personal Info (Username: WarriorNitz, Name: Nithiya K, Email: nithiya070399@gmail.com, Password, Bio: One of the Creators of Workshake, Location: Singapore, Skills: HTML, CSS, React.js, Node.js, Javascript, Java, Python), Social Info (Instagram, Facebook, Your Website, LinkedIn, YouTube, Twitter), and an 'Update' button.

Figure 5.3.2: Edit Profile

Users can go view profiles of other users and can follow them.

Figure 5.3.3: List of all users

For example, when user A follows user B, B's user object reference will be added to A's following array and A's user object reference will be stored in B's followers array. The picture below depicts how the data is stored in the database.

```

    < followers: Array
      0: ObjectId("60ac6c8a81ac1c31b025d46b")
      1: ObjectId("60ac6e69c7b5c728c80959ff")
      2: ObjectId("60b44cdcc74c5646ac49d860")
    < following: Array
      0: ObjectId("60ac6c8a81ac1c31b025d46b")
      1: ObjectId("60ac6e69c7b5c728c80959ff")
    > profilePictures: Array
      name: "Nithiya K"
  
```

Figure 5.3.4: Followers' Data stored in DB



Figure 5.3.5: Following and Followers - Flow diagram

The same procedure happens when B follows A back. Also when a user follows someone, they get a notification saying:

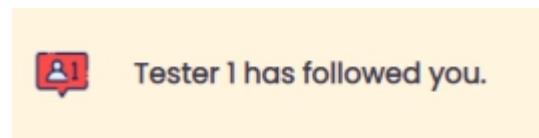


Figure 5.3.6: Following Notification

If both the users are following each other, the chat feature for them to text each other will be enabled.

Users can create, like and comment on posts. Text, image, video and YouTube posts are different kinds of posts that can be posted by a user.

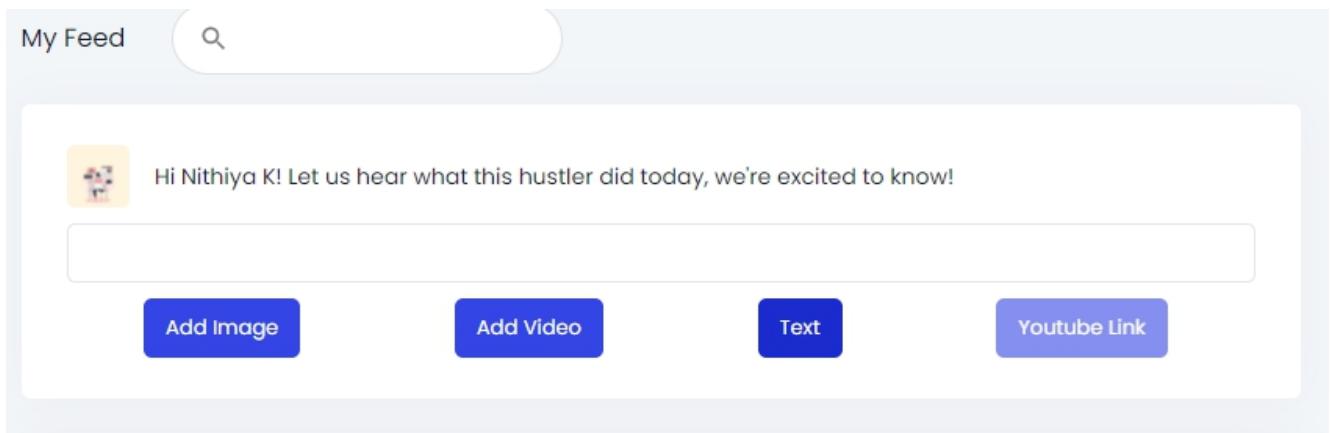


Figure 5.3.7: Home Page

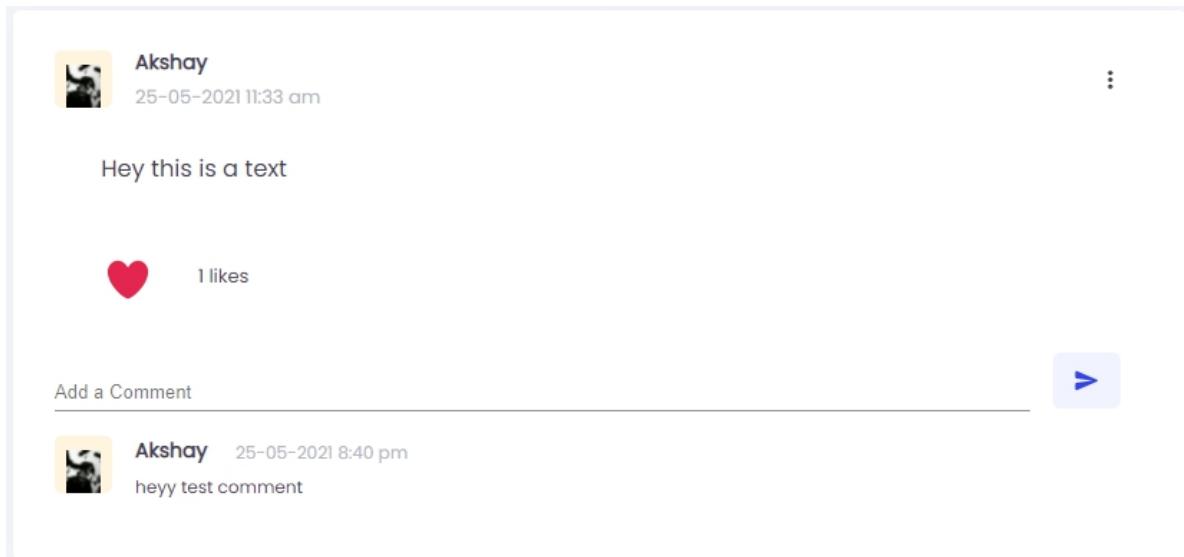


Figure 5.3.8 : Text Post

When a user creates a post, a document with a unique ID is created under the ‘posts’ collection. An example of a post document is shown below:

```
_id: ObjectId("60ac71180bb48433e0208ebe")
▼ reportCounter: Array
  0: ObjectId("60ac6c8a81ac1c31b025d46b")
▼ photo: Array
  created: 2021-05-25T03:33:48.901+00:00
▼ liked_by: Array
  0: ObjectId("60ac6ba8c9ec44408892a5df")
  postType: "text"
▼ tags: Array
  title: "Hey this is a text"
  postedBy: ObjectId("60ac6c8a81ac1c31b025d46b")
▼ comments: Array
  ▼ 0: Object
    PostedOn: 2021-05-25T12:40:17.152+00:00
    _id: ObjectId("60acf03e5ee14246e03df50b")
    comment: "heyy test comment"
    userId: ObjectId("60ac6c8a81ac1c31b025d46b")
    userName: "Akshay"
  __v: 3
```

Figure 5.3.9 : Post’s Data stored in DB

As seen above, the user object reference of the person who made the post is saved in the document. When a user likes the post, the user object reference is added to the ‘liked\_by’ array. Similarly, when a comment is posted, an object containing the time, the user object reference and the text is saved under the ‘comments’ array.

There is also a feature for users to report posts. When a post is reported, a warning in the form of notification will be sent to the user who posted the post.



Figure 5.3.10 : Alert Notification on report of post

When the same post is reported more than 3 times by different users, the website will automatically delete that particular post.

The images and videos posted by the user are stored in cloudinary. Cloudinary offers a cloud based image and video retrieval service.

#### 5.4. Chats :

In our application Real-Time Chat is powered by Socket.IO which is a web application that allows multiple users to have a private and public chat. By using this we can overcome the additional get request.

This chat application has been used in 2 different areas in our website:

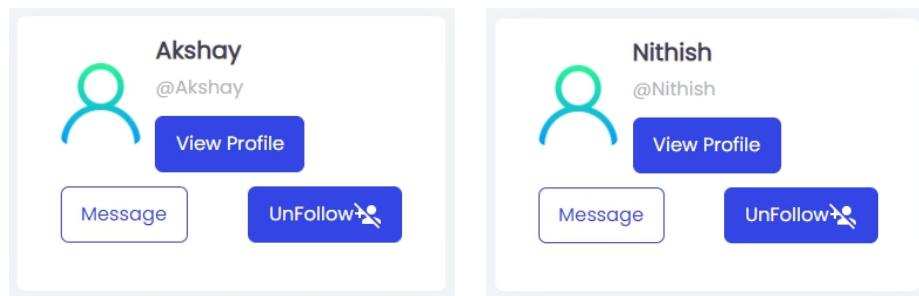
1. Individual chat – Upon mutual connection between users
2. Group chat – Upon creation of the team, all the users will be added to the group chat created for that project.

##### 5.4.1. Basic Working of Individual chat :

Once the user(1) follows another user(2) and in return user(2) follows user(1) he/she can individually chat with each other. If any of the users sends the message it will get collected in the common message pool named as "personal\_message" with user id of both recipient and sender. This is done by: socket.on("personal\_message", { sender id ,recipient id ,message ,created}).

Followed by which it gets emitted to the personal chat of the recipient by socket.emit("personal\_message" + recipient\_id). This emitted message will be tracked by socket.on("personal\_message" + recipient\_id) and will be displayed in the chat application.

For example: Let user1 be Akshay and user2 be Nithish. Both the users have to be mutually following to chat with each other



Perspective from Nithish's Profile

Perspective from Akshay's Profile

Figure 5.4.1.1 : Follow Users

Below are the pictures of chat between users Akshay and Nithish :

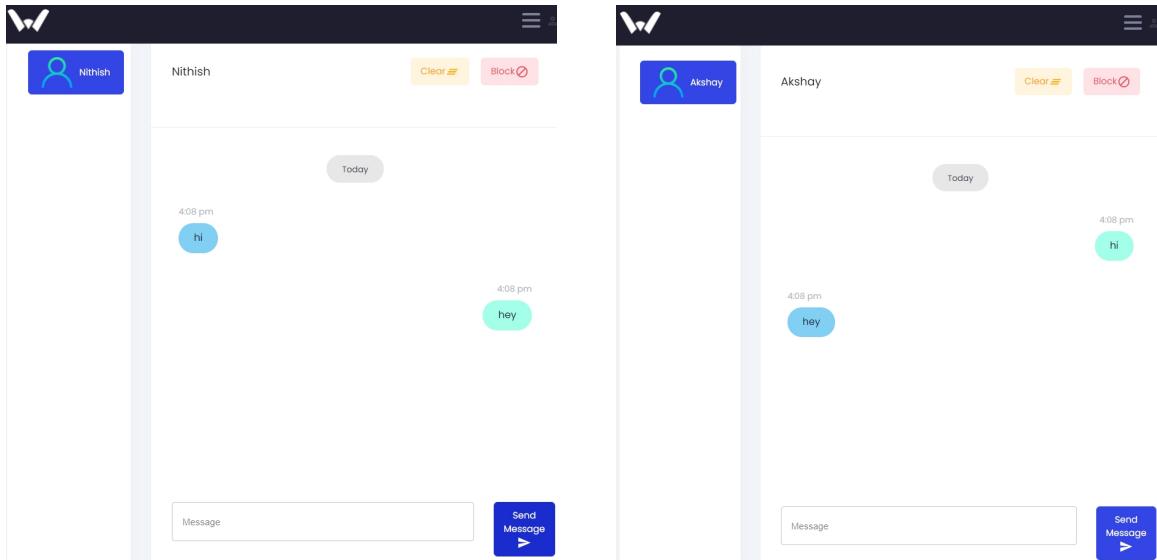


Figure 5.4.1.2 : Chat between 2 users

#### 5.4.2. Basic Working of Group Chat :

Once the project gets created, the team member will be added to that group chat of that project. If any of the team member sends the message, it will get collected in the common pool named as "message" with project\_id as follows :

```
socket.on("message", { name, message, created, project_id }).
```

This will then emit the message to that project by `socket.emit("message" + project_id, { name, message, created })`. If the validation becomes true, the message will be displayed to the respective project chat. ( Note : All the user Id and project Id are unique in nature. )

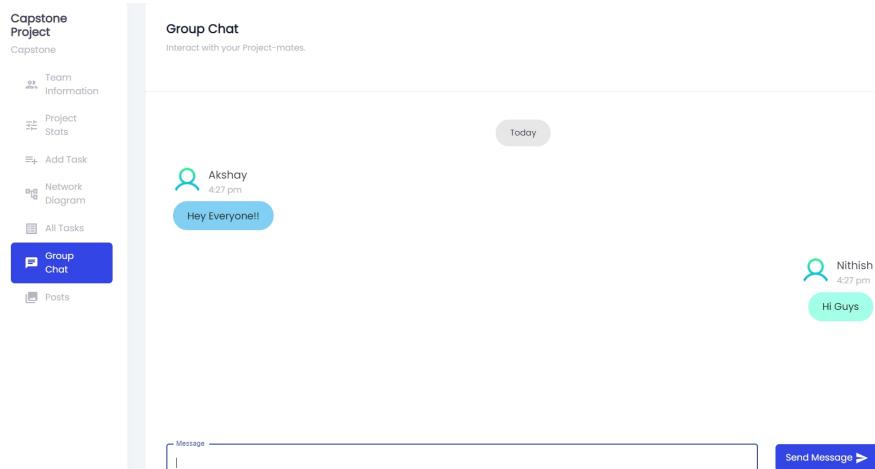
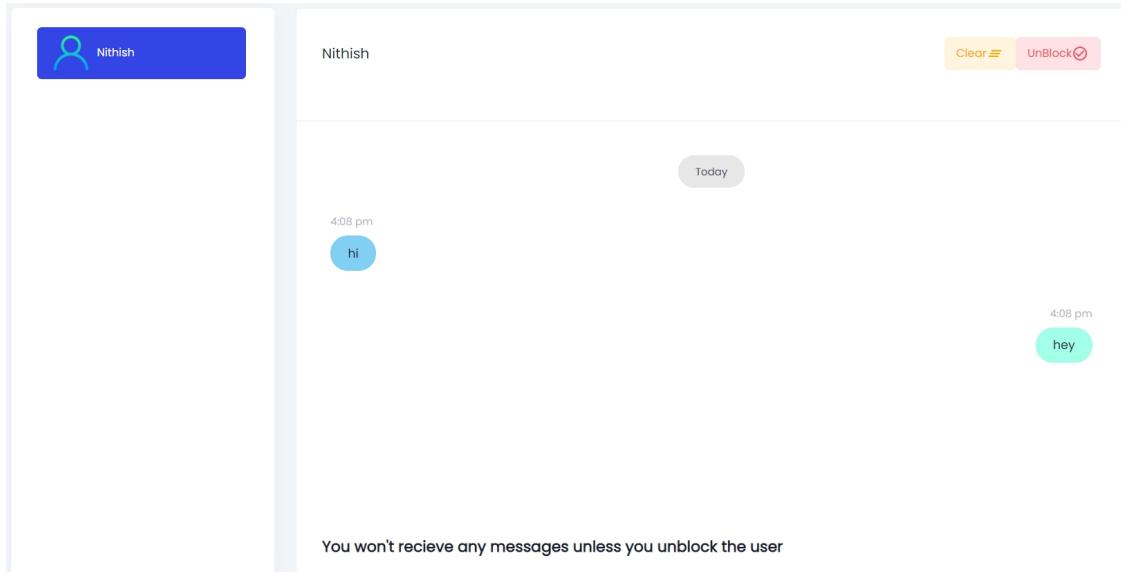


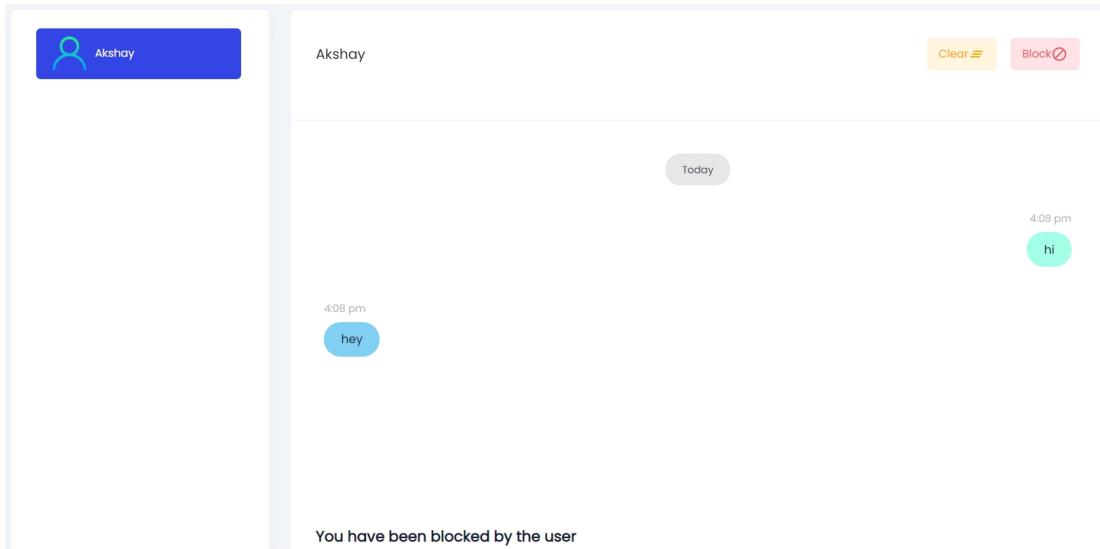
Figure 5.4.2.1: Group Chat

### 5.4.3. Chat Block :

If the user finds any of his/her followers annoying/spamming, we have provided an option to block that specific user so that the user won't receive any further messages from that user. In return the user won't be able to send a message to that user unless he/she unblocks that user.



Akshay blocked Nithish.



Nithish has been blocked by Akshay

*Figure 5.4.3.1 : Block Users*

## 6. Future Work

This project can be classified majorly into two different use cases:

In **Education Institutions**, with all classes being online, professors are being subjected to a lot more work than in school or college. The student's perspective also involves problems as there is a problem in the collaboration with other students so as to properly manage and monitor projects that are to be submitted online.

This Project can be implemented with a mentor login and student login for all the students involved in the institution and they can be monitored (by the mentor / professor) or monitor the project (checking colleague's progresses, inter relating work and managing time, so on..)

### Advantages:

- For Mentors/Professors:
  - Projects that are done under them can be instantly monitored upon to check the progress of it.
  - More detailed and Clear information of the project is given in the portal.
  - Time can be monitored properly, reviews can be more justified as well.
  - Mark allotment can be based on the promised tasks and completed ones.
  - Overdues can be monitored and marks can be allotted based on that
  - Potential projects can be identified for research paper work or further improvements.
  - Can be viewed at anytime, anywhere on their respective devices with just a login
- For Research Scholars:
  - A chance to find the right set of people to work with
  - Research papers can be made with these set of people
- For Students:
  - A chance to find other students who are best suited for working with them
  - Project tasks to be worked on for each review is clearly established
  - Tasks can be monitored upon by fellow members
  - Progress of the project can be viewed anytime
  - Planning and Slacking can be used to plan daily schedules

- Stress is reduced as the number of unpredictable details on a project is lessened
  - Similar projects already done on the website can be viewed upon
  - Other projects can be taken for reference and their team members can be contacted upon.
  - A Record of the projects made by them is created
  - The Digital Footprint made can be shown for Career advancements such as Jobs or Degrees
  
- For the Educational Institution:
  - The Institution can gain data on the projects that their students have done and that can help it analyze students, analyze strengths of an individual / group of students.
  - Institutions can also figure out what is trending in the college and if the topic seems to be new, Changes in syllabus or New subjects can be implemented for the betterment of students.
  - The institution can also stop redundancy of the projects that are present in the college. No senior to junior passing on of projects will be encouraged further.
  - Innovative candidates, capable of multiple skills and teamwork/leadership are built in the process of the course.

In the **real world scenario**, this project can be implemented as a live project that connects, people who are interested to make their idea real. It can be moulded into applications for iOS and Android, Admin panels can be made for easy access to data and properly managed to maintain and connect multiple people and projects. Under the consideration of a realistic scenario a base crowd is supposed to be built and the target audience is mainly college/school students , freelancers, people who have jobs but want to work on something new, Research scholars who would like to make new research papers or just anyone who would like to do something new. As we Consider this to be **Non Restricted** users gain access to **People from All Circles**, It increases their chances at Gaining a new ability, Drastically increases exposure, Knowledge improves, Current day projects are brought into sight, Trends can be seen, New ideas can be brought into existence and many more.

### **Advantages:**

- **Students** can gain Inter Institutional Contacts / Universal Contacts ( for eg: projects for a College student from VIT Vellore can collaborate with the **Right** person in VIT's other branches, IITs, NITs, BITS,etc.. And the possibilities are endless)
- **Entrepreneurs** who want to make a Start Up find the right partners.
- **Research Scholars** can gain the right set of people whom they can work with to produce a research paper.
- **Freelancers** can gain contacts to make a project with (maybe also get paid from the project)
- **Anyone.** can work on something they want to / something they love and also with the right team to make an Idea real.

## **7. References**

Journal:

1. Aznida Hayati Zakaria, Norizah Haron, and 6 others, "A Practical Approach using CPM/PERT for Certain Activity Times in Construction Parking Project", World Applied Sciences Journal 35 (7): 1180-1184, 2017 ISSN 1818-4952© IDOSI Publications, 2017 DOI: 10.5829/idosi.wasj.2017.1180.1184
2. Kholil, Muhammad, Bonitasari Nurul Alfa, and Madjumsyah Hariadi, "Scheduling of House Development Projects with CPM and PERT Method for Time Efficiency (Case Study: House Type 36)." IOP Conference Series: Earth and Environmental Science. Vol. 140. No. 1. IOP Publishing, 2018.
3. M. R. Bintang, K. Rossa Sungkono and R. Sarno, "Time and Cost Optimization in Feasibility Test of CCTV Project using CPM and PERT," 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019, pp. 678-683, doi: 10.1109/ICOIACT46704.2019.8938466.
4. Mrs. Ruchita Shrimali Vyas, 2013, Scheduling Project Management Using Crashing CPM Network to get Project completed on Time, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 02, Issue 02 (February 2013).
5. Yuniarti, Endang, and Emmidia Djonaedi, "Optimizing Production Time in Book Printing using PERT/CPM.", DOI: 10.5220/0009896301020107 In Proceedings of the 8th Annual Southeast Asian International Seminar (ASAIS 2019)

## APPENDIX

### APPENDIX A - PERT CPM CODE:

#### Pert and Cpm Calculation:

```
pertCalc = () => {
  let nodes = this.props.nodes.map((elem) => ({
    ...elem,
  }));
  let ids = [];
  this.props.connections.map((connection) => {
    if (!ids.includes(connection.source)) {
      ids.push(connection.source.toString());
    }
    if (!ids.includes(connection.target)) {
      ids.push(connection.target.toString());
    }
  });
  let newNodes = [];
  nodes.map((node) => {
    if (ids.includes(node.id)) newNodes.push(node);
  });

  let tasksObject = ids.includes("1")
    ? {
        1: {
          id: "1",
          mostLikelyTime: 0,
          optimisticTime: 0,
          pessimisticTime: 0,
          predecessors: [],
        },
      }
    : {};
  newNodes.map((elem) => {
    if (
      elem.data.predecessors.length === 0 ||
      elem.data.predecessors === undefined
    )
      return;
    elem.data.predecessors.map((pre, index) => {
      let id = this.getIdForObjectID(pre.toString());
      let predecessors = [...elem.data.predecessors];
      predecessors[index] = id.toString();
      elem.data = { ...elem.data, predecessors };
    });
  });
}
```

```

    });
  });

tasksObject = newNodes.map((elem) => {
  tasksObject[elem.id.toString()] = {
    id: elem.id.toString(),
    optimisticTime: elem.data.optimistic,
    mostLikelyTime: elem.data.time,
    pessimisticTime: elem.data.pessimistic,
    predecessors: elem.data.predecessors,
  };
  return tasksObject;
});

let tasksObjectFinal = tasksObject[tasksObject.length - 1];
let pert = {};
try {
  pert = jsPERT(tasksObjectFinal);
  this.props.setPert({ pert });
  console.log(this.props.pert);
  let slackObject = {};
  slackObject = newNodes.map((elem, index) => {
    if (index !== 0 && index !== 1) {
      console.log("index:", index);
      console.log("task:", elem.data.label);
      let created = elem.data.created;
      let createdDate = new Date(created);
      let earliestFinish = moment(createdDate, "DD-MM-YYYY").add(
        pert.earliestFinishTimes[index + 1],
        "days"
      );
      earliestFinish = earliestFinish._d;
      const diffTime = Math.abs(earliestFinish - createdDate);
      const duration = Math.ceil(diffTime / (1000 * 60 * 60 * 24));
      let today = new Date();
      let todayDate = new Date(today.toUTCString());
      let earliestStart = moment(createdDate, "DD-MM-YYYY").add(
        +pert.earliestStartTimes[index + 1],
        "days"
      );
      earliestStart = earliestStart._d;
      const diffTime2 = Math.abs(todayDate - earliestStart);
      let daysDone = Math.ceil(diffTime2 / (1000 * 60 * 60 * 24));
      let finaldiff = Math.abs(duration - daysDone);
      let days = Math.round(
        (earliestFinish - todayDate) / (1000 * 60 * 60 * 24)
      );
      slackObject[elem.id.toString()] = {
        id: elem.id.toString(),
        earliestFinish: earliestFinish,
        earliestStart: earliestStart,
        duration: duration,
        days: days,
        finaldiff: finaldiff,
        daysDone: daysDone,
        diffTime: diffTime,
        diffTime2: diffTime2,
        created: created,
        createdDate: createdDate,
        label: elem.data.label,
        type: elem.data.type,
        predecessors: elem.data.predecessors,
        successors: elem.data.successors,
      };
    }
  });
}

```

```

);
console.log(earliestStart, todayDate, earliestFinish);
console.log(
    "no. of days done:",
    Math.round(todayDate - earliestStart) / (1000 * 60 * 60 * 24)
);
console.log(
    "no. of days left:",
    Math.round((earliestFinish - todayDate) / (1000 * 60 * 60 * 24))
);
days = pert.slack[elem.id] !== 0 ? days + pert.slack[elem.id] : days;
if (days < 0) {
    elem.data.assignedTo.map((person, i) => {
        addToKickOutCounter(
            elem.key,
            person,
            this.props.project._id
        ).then((data) => {
            if (i === elem.data.assignedTo.length - 1) {
                console.log(data.result.overdueCounter);
                let overdues = data.result.overdueCounter;
                Object.keys(overdues).map((user) => {
                    if (overdues[user].length === 3) {
                        console.log("time to kick out +" + user);
                        this.props.notificationAdded({
                            userId: getCurrentUser()._id,
                            message: `3 hits, time to kick out ${user}`,
                            type: "KickOutUser",
                            userObjId: user,
                            project: this.props.project,
                        });
                        toast.error(`Warning : 3 overdues by ${user}`);
                    }
                });
            }
        });
    });
}
slackObject[elem.data.label] = {
    slack: pert.slack[elem.id],
    days: days,
    daysPassed: Math.round(
        (todayDate - earliestStart) / (1000 * 60 * 60 * 24)
    ),
}

```

```

        earliestStartDate: earliestStart,
        todayDate: todayDate,
        earliestFinishDate: earliestFinish,
        overdue: days >= 0 ? false : true,
    );
    return slackObject;
}
);
console.log("slacks Object:", slackObject[slackObject.length - 1]);
let obj = slackObject[slackObject.length - 1];
this.props.setSlacks({ slackObject: obj });
let newNodesObject = {};
newNodesObject = newNodes.map((node) => {
    newNodesObject[node.id] = node.data;
    return newNodesObject;
});
newNodesObject = newNodesObject[newNodesObject.length - 1];
let criticalPathData = {};
criticalPathData = pert.criticalPath.map((id) => {
    criticalPathData[id] = newNodesObject[id];
    return criticalPathData;
});
criticalPathData = criticalPathData[criticalPathData.length - 1];
console.log("criticalPathDataObject:", criticalPathData);
this.props.setCriticalPath({ criticalPath: criticalPathData });

this.props.setExpectedTime({
    expectedTime: Math.floor(this.props.pert.latestFinishTimes.__end),
});
putExpectedTime(
    this.props.project._id,
    Math.floor(this.props.pert.latestFinishTimes.__end)
);
} catch (err) {
    this.props.setPert({ pert: {} });
    this.props.setExpectedTime({
        expectedTime: 0,
    });
    putExpectedTime(this.props.project._id, 0);
}
};

```

### Connection On Network Diagram:

```
onConnect = (params) => {
  if (this.props.project.leader.toString() === getCurrentUser()._id) {
    let source = params.source;
    let target = params.target;
    if (source !== undefined && target !== undefined) {
      let edge = {
        id:
          "reactflow__edge-" +
          source.toString() +
          "null-" +
          target.toString() +
          "null",
        source: source.toString(),
        sourceHandle: null,
        target: target.toString(),
        targetHandle: null,
      };
      let sourceId = "";
      let targetId = "";
      this.props.nodes.map((elem) => {
        if (elem.id === source) {
          sourceId = elem.key;
        }
        if (elem.id === target) {
          targetId = elem.key;
        }
      });
      putPredecessors(this.props.project._id, targetId, sourceId).then(() => {
        console.log(sourceId + " has new Predecessor " + targetId);
      });
      putConnections(this.props.project._id, sourceId, targetId).then(() => {
        console.log("connection " + sourceId + "to " + targetId + "added");
      });
      this.props.connectionAdded({ connection: edge });
      this.pertCalc();
    }
  }
};
```

## APPENDIX B - Image Moderation:

Image Moderation and Upload:

```
exports.convertToWebp = (req, res) => {
  const cloudinary = require("cloudinary").v2;
  cloudinary.config({
    cloud_name: "*****",
    api_key: "*****",
    api_secret: "*****",
  });
  console.log("file: ", req.file);
  let file = req.file;
  console.log("path:", file.destination + file.filename);
  sharp(file.destination + file.filename)
    .resize(1280, 720)
    .webp()
    .toFile(
      file.destination + file.filename + " edited.webp",
      async (err, info) => {
        if (err) console.log(err);
        else {
          console.log(info);
          let obj = await checkImage(
            file.destination + file.filename + " edited.webp"
          );
          console.log(obj);
          if (obj.nudity >= 90 || obj.violence >= 90) {
            fs.unlink(
              file.destination + file.filename + " edited.webp",
              function (err) {
                if (err && err.code == "ENOENT") {
                  console.info("File doesn't exist, won't remove it.");
                } else if (err) {
                  console.error("Error occurred while trying to remove file");
                } else {
                  console.info(`removed`);
                }
              }
            );
            fs.unlink(file.destination + file.filename, function (err) {
              if (err && err.code == "ENOENT") {
                console.info("File doesn't exist, won't remove it.");
              } else if (err) {
                console.error("Error occurred while trying to remove file");
              }
            });
          }
        }
      });
};
```

```

        } else {
            console.info(`removed`);
        }
    });
    return res
        .status(200)
        .json({ message: "Inappropriate Content", values: obj });
} else {
    cloudinary.uploader.upload(
        file.destination + file.filename + " edited.webp",
        (err, result) => {
            if (err) {
                console.log("error:", err);
                return res.status(400).json({ err });
            }
            console.log("result:", result);
            fs.unlink(
                file.destination + file.filename + " edited.webp",
                function (err) {
                    if (err && err.code == "ENOENT") {
                        console.info("File doesn't exist, won't remove it.");
                    } else if (err) {
                        console.error(
                            "Error occurred while trying to remove file"
                        );
                    } else {
                        console.info(`removed`);
                    }
                }
            );
            fs.unlink(file.destination + file.filename, function (err) {
                if (err && err.code == "ENOENT") {
                    console.info("File doesn't exist, won't remove it.");
                } else if (err) {
                    console.error("Error occurred while trying to remove file");
                } else {
                    console.info(`removed`);
                }
            });
        });
}

return res.status(200).json({ result });
}
);
fs.unlink(file.destination + file.filename);

```

```
        fs.unlink(file.destination + file.filename + " edited.webp");
    }
}
);
};
```

## APPENDIX C - Video Moderation:

### Video Moderation and Upload:

```
exports.postVideo = async (req, res) => {
  const cloudinaryVideo = require("cloudinary").v2;
  cloudinaryVideo.config({
    cloud_name: "*****",
    api_key: "*****",
    api_secret: "*****",
  });
  console.log(req.file);
  let file = req.file;
  let path = file.destination + file.filename;
  await makeScreenshots(path);
  console.log("Path:", path);
  let finalArray = await checkVideo(path);
  await rimraf("./videoScreenshots", function () {
    console.log("done");
  });
  let result = finalCheck(finalArray);
  console.log(result);
  if (!result) {
    cloudinaryVideo.uploader.upload(
      path,
      {
        resource_type: "video",
        chunk_size: 6000000,
      },
      (err, result) => {
        if (err) {
          console.log("error:", err);
          return res.status(400).json({ err });
        }
        console.log("result:", result);
        fs.unlink(path, function (err) {
          if (err && err.code == "ENOENT") {
            console.info("File doesn't exist, won't remove it.");
          } else if (err) {
            console.error("Error occurred while trying to remove file");
          } else {
            console.info(`removed`);
          }
        });
      });
    return res.status(200).json({ result });
  }
};
```

```
        }
    );
} else {
    console.log("Video path:", path);
    fs.unlink(path, function (err) {
        if (err && err.code == "ENOENT") {
            console.info("File doesn't exist, won't remove it.");
        } else if (err) {
            console.error("Error occurred while trying to remove file");
        } else {
            console.info(`removed`);
        }
    });
    return res
        .status(200)
        .json({ error: "Inappropriate Content", obj: finalArray });
}
};
```

# Time Management and Optimization Technique for Efficient Project Management and Completion

by Nithish S, S Akshay Prassanna, K Nithiya Soundari 17bce0002,  
17bce0041, 17bce2244

---

**Submission date:** 05-Jun-2021 11:52AM (UTC+0530)

**Submission ID:** 1600804342

**File name:** 17BCE0041,\_17BCE2244,\_17BCE0002\_Capstone\_Review\_3\_Document.pdf (6.39M)

**Word count:** 14416

**Character count:** 71762

# Time Management and Optimization Technique for Efficient Project Management and Completion

---

## ORIGINALITY REPORT

---

<b>12%</b>	<b>10%</b>	<b>3%</b>	<b>7%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

---

## PRIMARY SOURCES

---

- |   |   |      |
|---|---|------|
| 1 | <a href="http://www.coursehero.com">www.coursehero.com</a><br>Internet Source   | 3%   |
| 2 | Submitted to VIT University<br>Student Paper  | 1 %  |
| 3 | Muhammad Ridho Bintang, Kelly Rossa Sungkono, Riyanarto Sarno. "Time and Cost Optimization in Feasibility Test of CCTV Project using CPM and PERT", 2019 International Conference on Information and Communications Technology (ICOIACT), 2019<br>Publication | 1 %  |
| 4 | <a href="http://www.bocasay.com">www.bocasay.com</a><br>Internet Source   | 1 %  |
| 5 | <a href="http://idosi.org">idosi.org</a><br>Internet Source   | <1 % |
| 6 | Submitted to Udayana University<br>Student Paper  | <1 % |
| 7 | <a href="http://www.scitepress.org">www.scitepress.org</a><br>Internet Source   | <1 % |

8	riptutorial.com Internet Source	<1 %
9	Submitted to Liverpool John Moores University Student Paper	<1 %
10	censorbib.nymity.ch Internet Source	<1 %
11	Submitted to University of Johannesburg Student Paper	<1 %
12	Submitted to Indian Institute of Information Technology, Allahabad Student Paper	<1 %
13	arxiv.org Internet Source	<1 %
14	www.semanticscholar.org Internet Source	<1 %
15	studiousguy.com Internet Source	<1 %
16	worldwidescience.org Internet Source	<1 %
17	Submitted to Colorado Technical University Online Student Paper	<1 %
18	Submitted to Sim University Student Paper	<1 %

19	<a href="http://www.scribd.com">www.scribd.com</a> Internet Source	<1 %
20	Muhammad Kholil, Bonitasari Nurul Alfa, Madjumsyah Hariadi. "Scheduling of House Development Projects with CPM and PERT Method for Time Efficiency (Case Study: House Type 36)", IOP Conference Series: Earth and Environmental Science, 2018 Publication	<1 %
21	<a href="http://www.ijert.org">www.ijert.org</a> Internet Source	<1 %
22	Submitted to TSU, University College Student Paper	<1 %
23	Submitted to Chandler Unified School District Student Paper	<1 %
24	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
25	<a href="http://ethesis.nitrkl.ac.in">ethesis.nitrkl.ac.in</a> Internet Source	<1 %
26	Submitted to Hoa Sen University Student Paper	<1 %
27	<a href="http://docplayer.net">docplayer.net</a> Internet Source	<1 %

- 28 Submitted to Higher Education Commission Pakistan <1 %  
Student Paper
- 
- 29 Submitted to Indian Institute of Technology, Madras <1 %  
Student Paper
- 
- 30 Submitted to Belhaven University <1 %  
Student Paper
- 
- 31 Cory Gackenheimer. "Node.js Recipes", Springer Nature, 2013 <1 %  
Publication
- 
- 32 [www.maas.edu.mm](http://www.maas.edu.mm) <1 %  
Internet Source
- 
- 33 Adnan M. Awad, Mohammed A. Shayib, Ahmad M. Dawagreh. "Large sample prediction intervals for future geometric mean a comparative study", Communications in Partial Differential Equations, 2010 <1 %  
Publication
- 
- 34 Submitted to National Institute of Business Management Sri Lanka <1 %  
Student Paper
- 
- 35 Submitted to Universidad Autónoma de Nuevo León <1 %  
Student Paper
- 
- 36 Submitted to University of Bahrain <1 %  
Student Paper

---

		<1 %
37	dlib-pr.mpier.mpg.de Internet Source	<1 %
38	gitlab.ecs.vuw.ac.nz Internet Source	<1 %
39	precog.iiitd.edu.in Internet Source	<1 %
40	www.slideshare.net Internet Source	<1 %
41	Na Li, Rajkumar Murugesan, Lin Li, Hao Zheng. "IDEAL: An Interactive De-Anonymization Learning System", 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), 2020 Publication	<1 %
42	strmech.kuciv.kyoto-u.ac.jp Internet Source	<1 %
43	www.bartleby.com Internet Source	<1 %

---

Exclude quotes      On  
Exclude bibliography      On

Exclude matches      < 10 words