

A Mini Project Report on

# **Simple Social Media Platform to connect with college peers (Connecture)**

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

**Bachelor of Engineering**

in

**Computer Engineering**

by

**Riddhi Narkar (19102003)  
Devansh Katheria (19102027)  
Shauryan Singh (19102004)  
Aditya Yadav (19102006)**

Under the Guidance of

**Dr. Pravin Adivarekar**



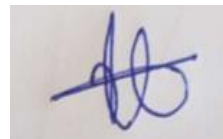
**Department of Computer Engineering**  
A.P. Shah Institute of Technology  
G.B.Road, Kasarvadavli, Thane(W), Mumbai-400615  
UNIVERSITY OF MUMBAI  
**Academic Year 2020-2021**

## Approval Sheet

This Mini Project Report entitled “**Simple Social Media platform to connect with college peers (Connecture)**” Submitted by “**Riddhi Narkar**” (19102003), “**Aditya Yadav**” (19102006), “**Devansh Katheria**” (19102027), “**Shauryan Singh**”(19102004) is approved for the partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering** from **University of Mumbai**.



Dr. Pravin Adivarekar  
Guide



Prof. Sachin H Malave  
Head Department of Computer Engineering

## CERTIFICATE

This is to certify that the mini project entitled “**Simple Social Media platform to connect with college peers (Connecture)**” submitted by “**Riddhi Narkar**” (19102003), “**Devansh Katheria**” (19102027), “**Shauryan Singh**” (19102004), “**Aditya Yadav**” (19102006), for the partial fulfillment of the requirement for award of a degree **Bachelor of Engineering** in **Computer Engineering** to the University of Mumbai, is a bonafide work carried out during academic year 2020-2021.



Dr. Pravin Adivarekar  
Guide



Prof. Sachin H Malave  
Head Department of Computer Engineering

Dr. Uttam D.Kolekar  
Principal

External Examiner(s)

1.

2.

Place: A.P. Shah Institute of Technology, Thane  
Date: 17/5/2021

## **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----  
(Signature)

-----  
(Riddhi Narkar 19102003)  
(Devansh Katheria 19102027)  
(Shauryan Singh 19102004)  
(Aditya Yadav 19102006)

Date: 17/5/2021

## **Abstract**

This project is based on the concept of making new and meaningful connections. By providing access to all college peers on a single platform, getting to know peers with a similar interest can help to guide and inspire students. We already have many sophisticated social platforms to connect with people, but, a setup at an institute-level would significantly help to bind everyone by leveraging the ability to connect digitally.

# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	Problem Definition . . . . .	11
1.2	Objectives . . . . .	11
1.3	Scope . . . . .	12
1.4	Existing System/Project . . . . .	12
<b>2</b>	<b>Literature review</b>	<b>13</b>
<b>3</b>	<b>Technology Stack</b>	<b>15</b>
<b>4</b>	<b>Benefits and Applications</b>	<b>18</b>
4.1	Benefits for society . . . . .	18
4.2	Benefits for environment . . . . .	18
4.3	Applications. . . . .	18
<b>5</b>	<b>Project Design</b>	<b>19</b>
5.1	Proposed System . . . . .	19
5.2	Flow of Modules. . . . .	20
5.3	Data Flow Diagram. . . . .	21
<b>6</b>	<b>Modules of System</b>	<b>22</b>
6.1	Landing page, Login, and Register. . . . .	22
6.2	User Profile. . . . .	24
6.3	Dashboard. . . . .	26
6.4	Navbar. . . . .	27
6.5	Posts. . . . .	27
<b>7</b>	<b>Project Implementation</b>	<b>28</b>
7.1	Actions. . . . .	28
7.2	Reducers. . . . .	29
7.3	Models . . . . .	29
7.4	Routes and API files. . . . .	30

# Contents

<b>8 Result and analysis</b>	<b>31</b>
<b>9 Annexure A</b>	
9.1 Gantt Chart. . . . .	32
<b>10 Future scope</b>	<b>33</b>
<b>11 Bibliography</b>	<b>35</b>
<b>12 Appendices</b>	<b>37</b>
Appendix-A:- Installation of node modules . . . . .	37

# List of Figures

3.1	Technology Stack . . . . .	15
5.1	Flow of modules . . . . .	20
5.2	Data Flow diagram . . . . .	21



# List of Tables

10.1	Gantt Chart . . . . .	32
------	-----------------------	----

# List of Abbreviations

SMP: Social Media Platform

# Chapter 1

## Introduction

Exploring new technology, knowledge, and skills is an important step to try to find a new passion or to polish an existing one. But, finding and connecting with new people who share a common interest can be a bit overwhelming at times.

We would be addressing this very issue by designing a centralized platform wherein students and teachers can build new connections, post information and stuff that can help to create an inspiring and encouraging environment to learn, develop and venture for all.

### 1.1 Problem Definition

For this mini-project, we aim to develop a SMP (Connecture) at an institute level wherein students can connect with and share tech-related ideas, knowledge and help foster a rich tech culture in our college.

### 1.2 Objectives

- 1) To develop a rich tech culture in college.
- 2) To help college students connect and interact in a way that would lead to their skill development.
- 3) To keep everyone updated about the current technical scenario and to present them with opportunities which would strengthen their vital skills in their college years.

## **1.3 Scope**

This project is will be implemented in an institute level. This way, all college students will have a dedicated and organized and centralized platform wherein they can find other students with similar interest and hence, can venture in the fields of their interest by helping, learning, and guiding each other. This will significantly boost their development curve, both when it comes to technical skills and soft skills alike. The underlying idea behind this project is to increase the quality and quantity of interactions students have about technology in an institute, which in turn is guaranteed to make students more open to opportunities to improve their skills, areas to do and learn from mistakes. The overall effect would be seen in the improving tech culture of the institute which is almost non-existent in 3 tier colleges.

## **1.4 Existing system/project**

When it comes to SMPs which are committed to provide people opportunities to connect, we really have many well-established sites which commit to provide powerful tools for connections. SMPs like Facebook really aims towards meeting new people and connecting with them to make a broader community. LinkedIn is focused on making meaningful connections with people who could help them in professional needs. But both these systems are implemented on a global level and their reach is huge. Bringing a well-planned mix of some of the features of these SMPs and changing the reach level makes this project different. We are leveraging the same – the concept of SMP to enhance connections, but on an institute level and in a way that benefits all.

## Chapter 2

# Literature review

We already have a lot of sophisticated SMPs in the market and which are in cut throat competition with each other, and building and making newer versions of themselves to cater to the audience a bit more every time, and to make the UX better.

For the development of this project, we looking into and studying a lot different SMPs to learn and analyze their features and how they market them and how to develop or improve their existing features, and also that how the people react to different versions and features of the same. Studying all these gave us a very clear idea how to design a social site which could actually help people. Also, how and which features could we add to Connecture, so that it would be better was a difficult question to start with, which was solved after carrying a detailed study of these social media sites.

Connecture could be thought of as an amalgamation of two might, and already well-established SMPs, namely LinkedIn and Facebook. It is a bit professional, and related to one's personal growth in their career or field of interest, thus, dropping in hints of LinkedIn type of SMP. On the other hand, Connecture also aims that new people meet each other and make new friends and connects, so the ease of doing so should also be included in Connecture, here is where Facebook comes into the picture.

Connecture is inherently a bit different than both of these SMPs. What makes it different, is that is not a wide spread social platform. It is special and different, as it would be deployed on an institute level. That is, only and only the students of a particular institute can make accounts on Connecture. This allows students to enhance their connections and find their connections and new friends in their college a lot easier, as they have a service wherein all the students are available with their respective profiles on Connecture.

Also, a lot of SMPs had this concept of “followers and following”, which is the “friends” on Facebook and the “connections” of LinkedIn. But we, did not include any of these “followers and following”, as this would decrease the quantity of connections on Connecture a lot. Also, it would hamper the process of meeting new people. So with a view to avoid all of that, we did not include any such system on Connecture.

# Chapter 3

## Technology Stack

### A) FRONT END

#### 1) HTML

HTML is responsible for the structuring and placement of the content. It covers the way content is organized and where it is positioned on the page in whatever case. It is the backbone.

#### 2) CSS

CSS is responsible for the representation of the content. It includes colors, fonts, background stuff, peculiarities of a layout, etc.

#### 3) REACT

React.js is the declarative JavaScript framework for creating dynamic client-side applications in HTML. Through React we can build up complex interfaces through simple Components, connect them to data our backend server, and render them as HTML. React's strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles expected from a modern web framework: great support for forms, error handling, events, lists, etc.

## **B) BACK END**

### 1) Node.js and Express.js

Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses. By making XMLHttpRequests (XHRs) or GETs or POSTs from React.js front-end, you can connect to Express.js functions that power the application. Those functions in turn use MongoDB’s Node.js drivers, either via call backs for using Promises, to access and update data in your MongoDB database.

### 2) MongoDB

MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model. The Application stores any data (user profiles, content, comments, uploads, events, etc.), JSON documents created React.js front end can be sent to the Express.js server, where they can be processed and (assuming they’re valid) stored directly in MongoDB for later retrieval.



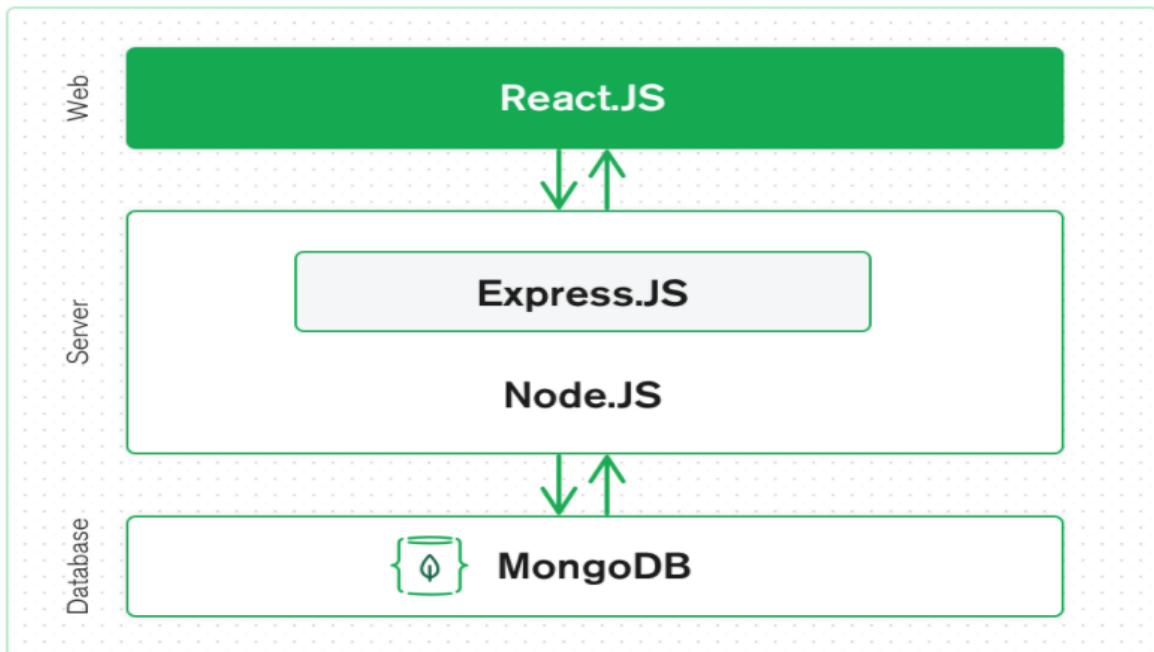


Figure 3.1: Technology Stack

# Chapter 4

## Benefits and Applications

### 4.1 Benefits for society

This initiative can help students to be able to form strong communities which in turn can help students of any institute to progress. It provides a fair opportunity and a seamless service for all students to excel in their career and to find people with similar interests and goals. Being able to find such a supportive company keeps a healthy competition alive between students and a collaborative and supportive environment is created. By doing this, college students can not only make new friends, but also meet new mentors, guides, or juniors with similar mindset and help them carve their way. This can build a tech intensive ecosystem.

### 4.2 Benefits for environment

This project will have a dark theme UI which would use less power. Also due to its simple nature, it's working is also not as complicated as other alternatives in the market, which again uses less memory and power.

### 4.3 Applications

- 1) Will create a very supportive and collaborative environment for students to create communities.
- 2) Will help them to meet new people with similar interests so that they can work together to achieve a similar goal.
- 3) Can act as a medium to keep students updated with the current technical scenario.

# Chapter 5

## Project Design

### 5.1 Proposed System

At first, user will be prompted to create an account with some basic personal information like name, photo, year, areas of interest and contact details. After creation of an account, the user could sign in with the username and password. The profile of every user will be set as public for better and faster connections. There will be an option of posting pictures or just a text or even a question to interact with others and setup a connection. As the account would be public, so the posts made by anyone could be visible in everyone's feed and they can interact with it and posts could be liked and commented also. To sort the posts out, there will be tags regarding the content, if he/she wanted to use a tag they can easily do so. User will receive notifications when the feed is uploaded or if there are interactions with their posts.

## 5.2 Flow of modules

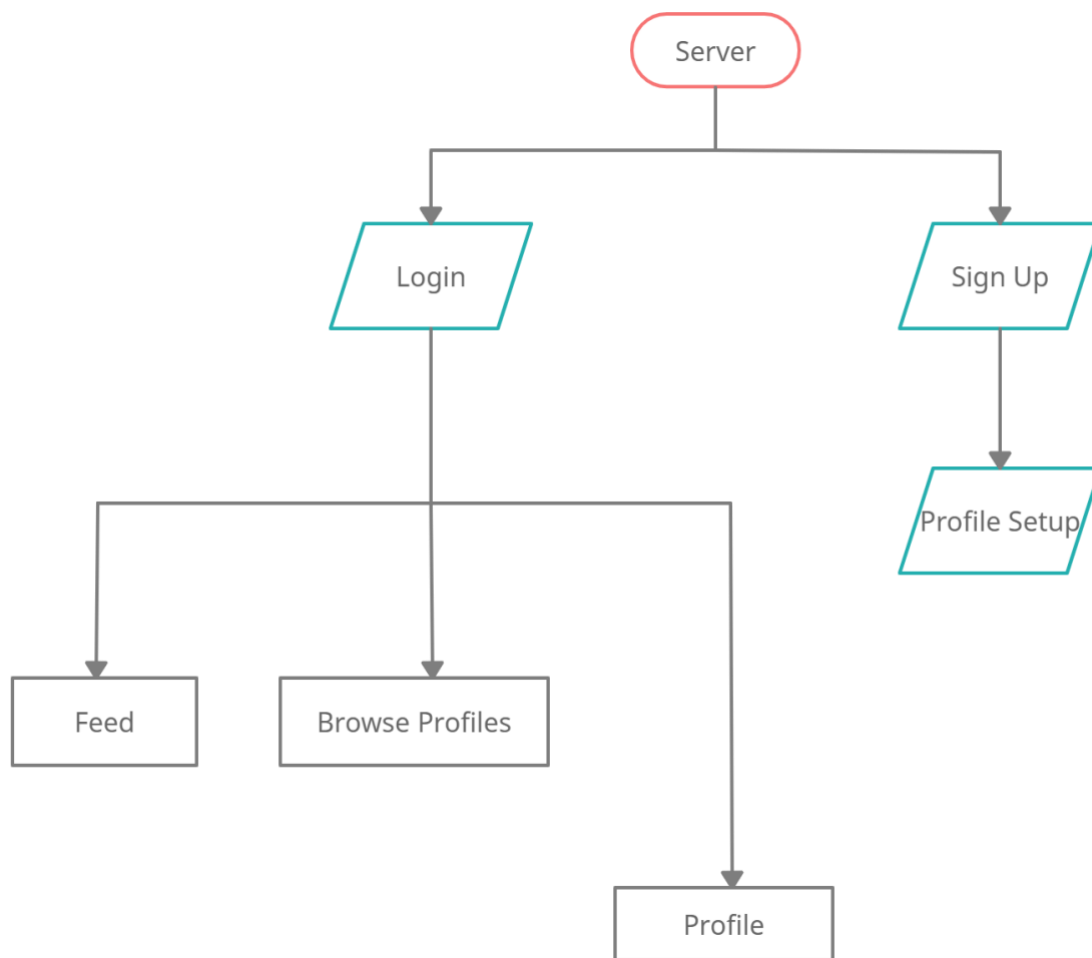


Diagram 5.1: Flow of modules

## 5.3 Data Flow Diagram

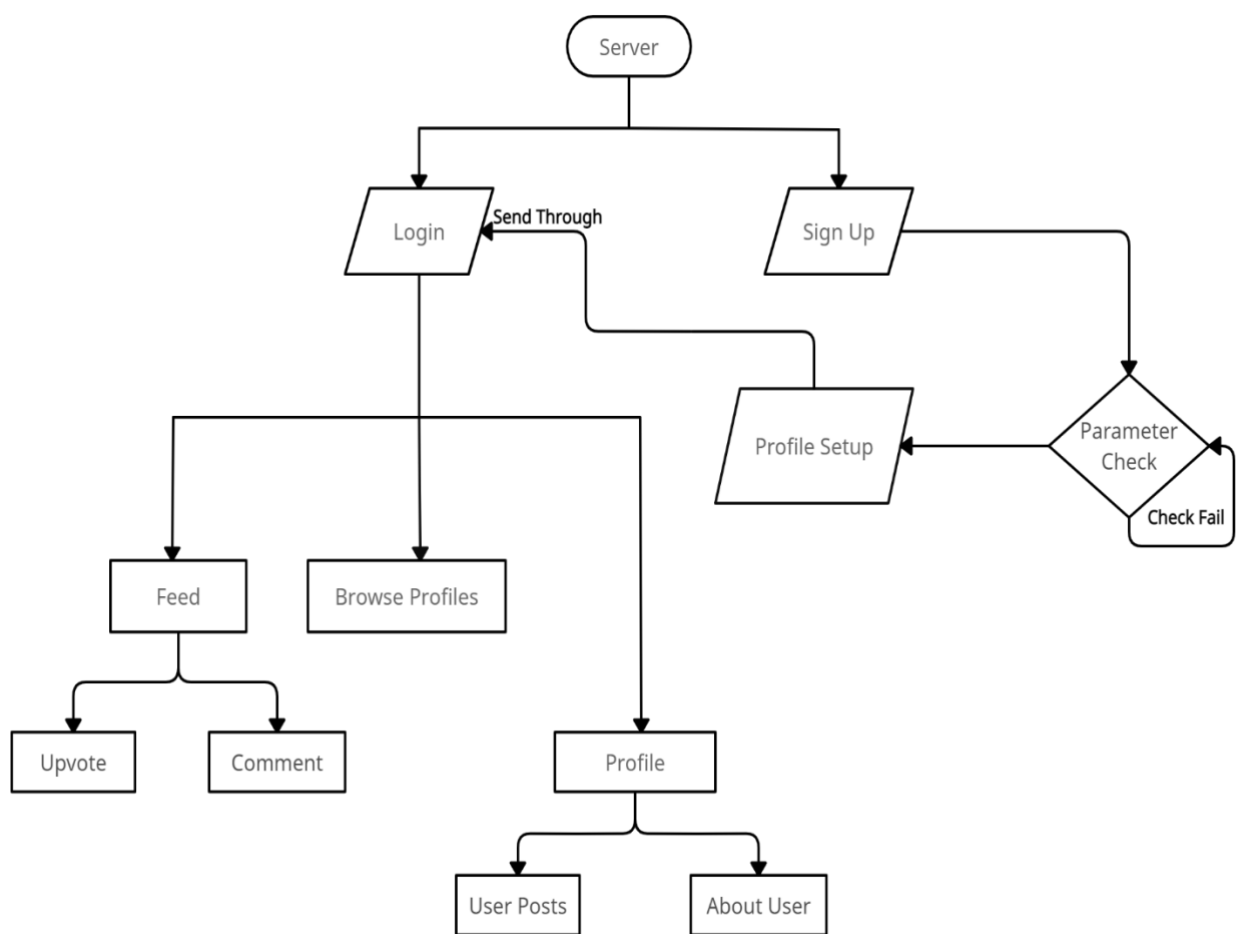


Diagram 5.2: Data flow diagram

# Chapter 6

## Modules of System

### 6.1.1 Landing page

When a new user visits our website for the first time, or visits the website after he has logged out from his previous session, he is taken to the landing page of the website. This page basically is title page of the web service. It includes the website name and a general introduction to the usage and aim of the website. Furthermore, the user can visit two different pages from this page, depending on the action he wants to perform. Either he can choose to login to his account, in case he already is a registered user of the website, or in case of a new user, he would like to register himself first, through the sign in button.

### 6.1.2 Sign up page

All new users, or first time users must create their accounts in order to use the website. They need to provide their name, email id, and a password for their account. They will be required to re-enter their password to confirm it, and then after hitting the submit button, an account with these credentials will be made in the backend.

### **6.1.3 Login in page**

Already registered users need to provide their password and email id while trying to login. The backend then authenticates the corresponding password to its corresponding email id and checks whether or not is valid.

If the password is valid, the user would be taken to his dashboard where he can see his past activity and feed.

If the password is invalid, then the user would be prompted to re-enter the correct password again.

In both these scenarios, that is for login and sign up, the user can anytime navigate to the last page, that is the landing page, and choose a new option (either login or sign up). Hence to avoid redundancy, we have removed the navbar from all these these pages, as it serves the same purpose.

## 6.2 User profile

Once a user has registered for a Connecture account, he may like to set up his Connecture profile. Since this is a very important part of UX, as this is how the user wishes to express himself on the site, we made sure that his experience involves the least amount of inconvenience. Hence, to improve the overall UX of the website, we did not ask for any additional details during the registration part. We wanted to make sure that any user is able to make an account in the least possible time and with less hassle. Hence we kept the more time intensive profile setting setup task later, after a user has a secure account. This way, the user can come back to the website via login and set up his profile anytime he wants. He can have an account, and his profile would be made with his minimum details, hence, making the process a bit more easier.

For the user profile, we ask the user's educational status. This website is for college students, so the possible values could be either 1<sup>st</sup> year, 2<sup>nd</sup> year, 3<sup>rd</sup> year, or 4<sup>th</sup> year. But with a view to increase and enhance the quality and experience of the website, we made a space for an institute's alumni too in the education status. A user, after passing out from college can still be a part of this community and to specify that, we can have an additional special educational status of "alumni".

This service is more or less, talking about technology, so the next necessary field we take in is the skills. These are the technologies or other sectors where the users are skilled or are learning. This gives other users an idea about what technologies or skills a particular user is interested in, which would help more in justifying the aim of the website.



We also ask for the GitHub username, as this would be a direct reflection of what projects the current user is working on. So this would be hassle free for user to just provide their GitHub links instead of manually stating their project details.

An short bio would also help a user to express himself in his own words and let his peers know what they need to know about him to connect further and make progress.

Finally, as an add on to more efficient connecting, a user can also provide his LinkedIn and Twitter links which further enhance connections.

After logging in, a user can also browse all other profiles on Connecture, which further help to support the idea of having a centralized system for peers.

## 6.3 Dashboard

There are different parts of Connecture, and to transition between them, the user often lands on dashboard. In react, Dashboard has different React pages in Connecture, one is the guide the user to set up his profile when he registers for Connecture. The next is when a user wants to edit his profile.

These are the transition pages between different forms and state of the dashboard component.

The first page is when the user is prompted to enter his or hers details on the his or her Connecture account. To enhance UX, we kept the action of setting up a profile and creating a profile different, so that any user can quickly create his or her profile. When a user does so, he would be required to first set up his profile, he can edit additional details into his profile, and his education credentials, which further enhance his or her profile.

The next dashboard page is actually the education form page. When the users wants to add his or her educational history and details, he or she will need to provide his or her school name, years for how much he or she studied there. A user can also add his or her college here and mark it as “current” educational institute. Since this is an institute level application, all the users will have this same info on their page, but this makes sense while differentiating the alumni users of the platform.

## 6.4 Navbar

A navigation panel or a navbar comes handy to navigate through a website with many pages. Connecture has a navbar which would be displayed once the user has logged in or signed in the website. Once entered, the navigation panel sits on the extreme top. It has links to other pages on the website such as the landing main page of Connecture, dashboard, user profiles, register, and login.

This navigation pane is also available on the landing and login page for a better UX. When clicked on “Connecture”, it takes you to the landing page of the website.

## 6.5 Posts

Posts, again, are implemented as react forms on Connecture. Any user can make a post and publish it on Connecture. Since, Connecture doesn't have the “followers and following” system, there is no way to know what post to show which user. This in fact, enhances the usage of Connecture as all the members can view all the posts made by any user.

# Chapter 7

## Project Implementation

### 1) **Actions**

Actions are those files in React which are responsible for carrying out specific tasks. They are the only files which provide information to the store file of the application. They include a type value in them. They are instructions on what to do when a specific state is obtained. The reducers, use these action files to do the state changes. Together, in React, we can thus use actions and reducers to make changes in component states.

Auth action has code which involves how to authorize any user, and what to do when a user is authorized or when a user isn't authorized. This gets called when a new user logs into Connecture.

Profile action file includes the instructions for when users create and set his or own profile. We have included create profile, add education, get profile items, and get profiles under a single code for profile. It contains all the necessary actions related to profile.

## 2) Reducers

Reducers are an amazing part of React and one of the reasons why it makes things simpler. These are the only way in React to change states. It takes in the state and the action and gives a new state in the output. Basically, the action tells what to do for a particular state and reducer knows how to do that so the state changes.

Auth reducer makes the state change from not logged in to logged in. As the features are different once you log in, like you have access to your profiles, etc, a logged in state is a different state than a logged out state.

Then we have the profile reducer. This is used when a user wants to change and update his profile. So the old profile will be the old state and the edited profile will be the new state.

## 3) Models

For the database, in MERN stack, we use MongoDB. Then, in React file structure, we have some model files which represent the data which user can interact with.

We can have models designed for various states, so it contains all the fields of data that a particular state owns and uses.

Under post model, we have the extensive set of attributes which a post can possess. So it includes some details about the user who made the post, and details like the number of likes, and which user liked that post, or comments, and who made that comment under what name(user name), and the date when the post was made.

For the profile model, we have all the data which a profile stores, so all the credentials like, name, year of study, education details, skills, bio, github username, and social account fields.

For the user model, we basically contain the login credential of the user, like his username and password and email. The profile model, though related to a user model, (as a single user will have a separate instance of profile and user), is kept different.

## **4) Routes and API files**

Routes are some of the important parts of React's file structure. It is responsible for rendering the UI if the current locations matches the route path. They are so important because they keep the URL and the UI in sync, which is important for any application to work properly and displays its states as and when the users interacts with the application.

API files make requests to the database to get data. Then can then be used to carry out some tasks. They are important, as they connect the endpoints together. The frontend gets data from the backend and hence actions could be performed likewise.

Auth api checked for the validity of the login, if not successful then it sends a error message. It finds the entered email and check for its corresponding password to validate the user trying to login into his or her Connecture account.

Profile api has various functions for different activities related to profiles. When a user sets up his profile, the backend needs to get updated, and when the user wants to edit his profile, the same data needs to fetched and updated from the backend. Similarly for deleting an account, we need to remove its database, so all these endpoints are programmed here.

User api is used when we need to see the details of the users. In Connecture, all the profiles are open to everyone, so when someone wants to see a other user's profile, this is used.

## Chapter 8

# Result and Analysis

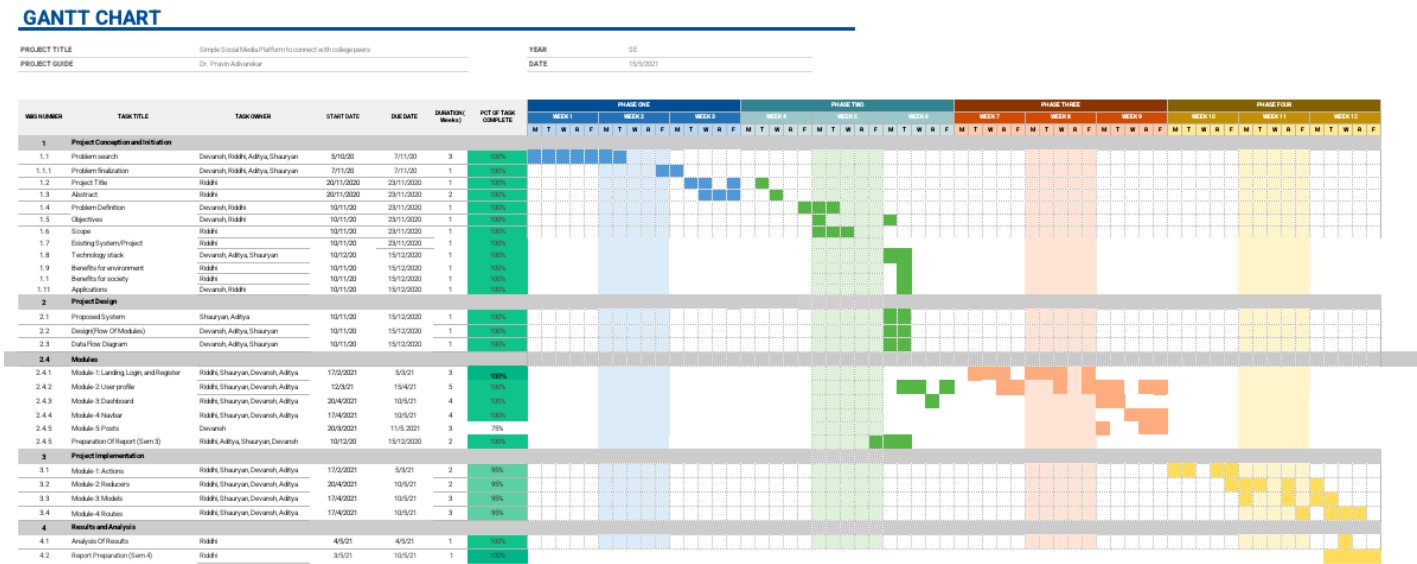
Connecture, thus being a social media platform, designed especially for college students at their own institute's level can significantly enhance the total experience with which peers interact with each other and find people with similar interests. Such platforms, which allows such powerful student interaction with each other are known to help with the tech culture of a college. A lot of people get benefitted by guidance, and get exposed to a lot of opportunities. The best part about such a system is that it is completely immune to a situation similar to the current pandemic, as this is totally an online system and still and help students to keep hope and make new friends and resume learning skills and technologies which they are passionate about or which they like.

Such a website also helps in making the common knowledge of a tech group wider. Facts, news and other events could be circulated and shared with a very good level of efficiency using such a service. Such fast sharing of information makes the students aware of the technical scenario of the industry, and as alumni are also a part of this service, they can provide their invaluable input in the form in industry related facts, so that students get more exposure and connect with their seniors too.

Lastly, Connecture can have a strong impact on the learning and development of a student with the help of his peers and help him find his passion and excel in his field with making and build and maintaining a strong tech culture supported by healthy connections.

## 10.1 Gantt Chart

23





# Chapter 10

## Future scope

Implementing a social media site can have endless possibilities to improve on. The following are some gray areas in our overall project where improvements are needed to make Connecture better at achieving the goal it was developed for.

In this model which we presented, a major drawback is that students cannot directly connect with each other via Connecture, but only through other contact details provided by the user. This counts against the UX which we are trying to build and achieve using Connecture. As this application aimed at improving and possibly revolutionizing the way students interact with each other given the scope of their institute and improving the overall tech culture of lower tier colleges, one possible solution for that would be to introduce chats in this service as a new feature. An inbuilt chatting option removes the reliability of the user of having to try to connect to other people via other already well-established connection platforms. This would not only make it easier and faster to connect with students, but would also mark a step in making Connecture independent of its reliability on other social platforms.

A chat feature like this comes with a lot of added features. When introducing a chat feature, it should be noted that our application doesn't have the concept of followers or following, again which is aimed towards better and transparent connections. But this would be a huge challenge to manage the privacy concerns over the chatting feature. For this, we could add a block feature if a user does not wish to continue any conversation with another user. Also,

some other additional features like reporting a chat could help us to maintain a certain level of decorum on Connecture.

Also, we could try to add notifications on this application, if someone tries to interact with a user's posts, then he or she might actually get a notification about the same. Or if any new post is been added to everyone's feed would also notify all the users of Connecture.

To give all these notifications a place to live, we might need to make some major changes to the current UI of Connecture. We can add a home page for every user where we can have a notification panel, so that the notifications are also easily accessible to the user. Or else, we can have an option available for the user, and only when clicking it, would make the notification panel to appear, and then re-clicking would make it collapse.

In addition to this, we can have a greater level of UX, if Connecture caters to all possible expectations of the users. Like, we can have a new feature totally indigenous to Connecture called a "meet request". We could think of this like a friend request on Facebook. Instead of trying to add a friend, here on Connecture, a meet request would actually notify the user, telling him or her, that the other person, who is the meet request sender, is interested in meeting them and getting to know more about their work or need some guidance or help. This would make the process of asking out for help a lot easier, efficient and faster. This would also help in making Connecture more useful for a lot of students.

# Chapter 11

## Bibliography

- 1) ReactJS, “ReactJs’s Official Documentation: Getting Started”,  
<https://reactjs.org/docs/getting-started.html>
- 2) Express.js, “Express’s Official Documentation: express()”, <https://expressjs.com/en/api.html>
- 3) MongoDB, “MongoDB’s Official Documentation”, <https://docs.mongodb.com/>
- 4) Codevolution, “ReactJS Tutorial for Beginners”, YouTube playlist for beginners learning React  
  
<https://www.youtube.com/playlist?list=PLC3y8-rFHvwgg3vaYJgHGnModB54rxOk3>
- 5) Aman Dhattarwal: Apni Kaksha, “World’s most premium web development course: From beginners to Advance level”, YouTube playlist for learning web development  
  
[https://www.youtube.com/watch?v=l1EssrLxt7E&list=PLfqMhTWNBT3H6c9OGXb5\\_6wcc1Mca52n](https://www.youtube.com/watch?v=l1EssrLxt7E&list=PLfqMhTWNBT3H6c9OGXb5_6wcc1Mca52n)
- 6) Material Design, “A guide to dark theme UI and its colour scheme”,  
<https://material.io/design/color/dark-theme.html#properties>
- 7) W3Schools, “CSS and Flexbox”, <https://www.w3schools.com/css/default.asp>
- 8) W3Schools, “Styling React using CSS”, [https://www.w3schools.com/react/react\\_css.asp](https://www.w3schools.com/react/react_css.asp)
- 9) Medium, “Four ways to style React components” Author-Agath Krzywda, Medium Group: Codeburst.io, <https://codeburst.io/4-four-ways-to-style-react-components-ac6f323da822>
- 10) Sarah L. Fossheim, “How to add a gradient overlay to text with CSS”,  
<https://fossheim.io/writing/posts/css-text-gradient/>
- 11) Google Fonts, <https://fonts.google.com>

- 12) Stack Overflow, “How to use Google fonts in React.js?”,  
<https://stackoverflow.com/questions/40769551/how-to-use-google-fonts-in-react-js>
- 13) Flavio Copes, “Validating input in Express using express-validator”,  
<https://flaviocopes.com/express-validate-input/>
- 14) C# Corner, “CRUD operations using React, Nodejs, Express, MongoDB”,  
<https://www.c-sharpcorner.com/article/crud-operation-using-reactnodejsexpress/#:~:text=Node%2CMongodb%2CReact%20js%2C,Step%20by%20Step%20write%20code.&text=Create%20New%20Folder%2C%20ReactCRUD%2C%20if,folder%20after%20running%20this%20command>
- 15) zalmoxisus - GitHub, “Redux Dev tool extension”, <https://github.com/zalmoxisus/redux-devtools-extension>
- 16) {z} Koder, “React Redux Login, Logout, Registration example with hooks”,  
<https://github.com/zalmoxisus/redux-devtools-extension>

# Chapter 12

## Appendices

### Installation of node modules:

The technology stack used in this project is the MERN stack, which has some dependencies. To run this code, your system would need node modules, and here's how you can install them and run this project.

#### **Steps:**

- 1) Download the latest version of Node.js corresponding to your OS.
- 2) Go to the root folder of the project and run the following commands in your terminal:

```
~ npm install
```

This will install node modules in your root folder.

```
~ cd client
```

```
~ npm install
```

This will install node modules in your client folder.

- 3) To run React, run the command

```
~ npm run client
```

- 4) To run Express, run the command

```
~ npm run server
```

5) To run both, run the command

*~ npm run dev*

If you have installed node modules correctly, and have run these commands successfully, then React will run this website on your local server.

For any other guidance or support, check out these official websites of the libraries and other software we used to develop Connecture.

1) [NPM](#)

2) [Node.js](#)

3) [React.js](#)

4) [Express.js](#)

## **Acknowledgement**

We have great pleasure in presenting the mini project report on **Simple Social platform to connect with college peers(Connecture)**. We take this opportunity to express our sincere thanks towards our guide **Dr. Pravin Adivarekar**, Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Sachin Malave** Head of Department, Computer Engineering, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Student Name1: Riddhi Narkar**

**Student ID1: 19102003**

**Student Name2: Devansh Katheria**

**Student ID2: 19102027**

**Student Name3: Shauryan Singh**

**Student ID3: 19102004**

**Student Name4: Aditya Yadav**

**Student ID4: 19102006**

