A REPORT
ON

SOCIAL MEDIA APPLICATION USING REACT JS

Submitted by,

Mr. Dhanush Gowda B M - 20211ISE0015

Under the guidance of,

M Pushpalatha

Assistant Professor of School of Computer Science and Engineering

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship report **"Social Media application using react JS"** being submitted by "**Dhanush Gowda B M**" bearing roll number "**20211ISE0015**" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Science and Engineering is a bonafide work carried out under my supervision.

Ms. M Pushpalatha
ASSISTANT PROFESSOR
PSCS / PSIS
Presidency University

Dr. PALLAVI R
PROFESSOR &HoD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled "**SOCIAL MEDIA APPLICATION USING REACT JS "** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Information Science and Engineering**, is a record of my own investigations carried under the guidance of **M Pushpalatha, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NO	SIGNATURE
DHANUSH GOWDA B M	20211ISE0015	

INTERNSHIP COMPLETION CERTIFICATE

inventeron
nothing but smart..

Reg. No. : AAB 9565

Date : 10th April 2025

Internship Completion Certificate

This is to certify that **Mr.Dhanush Gowda B M (Inv1564) was** employed as an intern with our company **Inventeron Technologies And Business Solutions LLP** and successfully completed **15 weeks** of an Internship on the domain **Full Stack Web Development** from **10/02/2025 to 12/05/2025.**

We found him independent, motivated,duty bound,and a highly committed team player with strong conceptual knowledge.

We extend our heartfelt Congratulation on completing this internship successfully .We at Inventeron Technologies wish him all the success in his future endeavors

Thank you

For Inventeron Technologies And Business Solutions LLP

Managing Director

Inventeron Technologies and Business Solutions LLP

No.364, 19th Main Road,1st Block, Rajajinagar, Bengaluru - 560 010

+91 8660723524
+91 7353822642

hr@inventeron.com
info@inventeron.com

ABSTRACT

There are so many advancements in the field of technology and as the environment around us grows, we have to grow with it as well. Our idea of modernization in technology leads us to design something which will lead to advancement in social media application use.

Our project idea was to implement a social media app by using technology - React.js. The project is just a basic implementation of that of a social media app which you can identify by the way it looks. Our work was based around implementation and UI design and now we are trying to get this Idea towards finalization.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this internship on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the Internship.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and **Dr.Pallavi R** , Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this internship successfully.

We are greatly indebted to our guide **Ms. M Pushpalatha, Asst. prof** and Reviewer **Dr.Sampath A K,** Presidency School of Computer Science and Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this Internship.

DHANUSH GOWDA B M

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The social media applications in itself have Billions of users. It uses fundamental concepts like user authentication, Database management, UI/ UX design, and software development while taking it to an advanced level. The aim of this project is to Generally recreate a social media app which provides similar functionalities to the apps like Facebook, Instagram application while trying to improve on some of the aspects from a developer's point of view.

The project will utilize React.js and Material UI for development and Google Firebase for both Authentication and Database management. Using the web application, you can share images on your feed or story for a specified period of time after which the story vanishes. You can see the usernames of people who have seen your story by checking the story section. It's easy to scroll through the list if the list is short but now, imagine doing that with 10K followers, or even 200 followers. After posting a story, the long process of scrolling to see who viewed your story later begins.

There should be an alternative for the process and that is why we suggest adding a search bar in the story section so that a user can easily navigate through their stories leading to a better user experience.

Reasons to use ReactJs:-

1 A simple way to think about React is this: you tell React how you want your app to look when your data changes. React works and makes the change happen- updating your UI.
2 With changes in data, it automatically refreshes with it and it knows to only update the changed parts and does not refresh the whole UI. This improves the overall performance.
3 With React, we can create and reuse components. Therefore we majorly build components by the help of react. Since they are encapsulated components, their code reuse, testing, and debugging or removal of faults become easier.
4 Simple, declarative, and reusable. These are the top 3 major qualities of React.

1.1 Problem Statement

We all see modernization playing an important role in all of our mundane activities and as everything moves forward towards it, so should we and our applications. We have seen that after a certain period a new way of doing things takes over the previous one because of the benefits and better results of that new method. We are doing the same thing here. The social media applications currently in use have a good retrieval time but the functionalities can for sure be implemented in a better way and there are certainly more optimal solution. We are trying to create a different approach to using social media applications and their story feature, the idea is to implement a social media application and then when a person sees the story and it shows how many users have viewed the story we have a search bar present right there to help with the name search if we want to search for a specific person in that list, which takes into account the already existing resources and their utilization of user data. The technologies which will be the building base of our social media app will be ReactJsand we try to implement a social media app with better story functionality while not dropping out on the user experience. The end goal is to create a working prototype for the project in order to show what and how exactly we are making this application work and how it takes our data into account.

Chapter 2

LITERATURE SURVEY

In an article titled "The Future of Open Source Software," author Junli Xu classifies Android as an open-source operating system—quite different from previous closed-source operating systems. The author further explains that with an open-source OS, we certainly have the opportunity to work on whatever we want for ourselves. Subsequently, the author gets specific regarding the music player and what it will do. According to the writer, although users like to have UIs that are pretty and colorful it certainly takes the mobile device for more resource utilization. The writer says they are composing a media player for android which has certainly less resource utilization while being simple, convenient and user-friendly as well. Using JAVA as their main language and eclipse as IDE is the method of implementation. This is how the author creates an Android app with playlists, an interface, menus, and playback settings.

According to **Ravi Kumar, Numer Mulla, Dishank Gangar,** the authors have given us an idea of how current recommendation systems are inefficient and most of the recommendations are just based on the user's behaviour on similarity between songs.

Therefore, the creators have made up their minds to develop a music application named as Illusion which takes into account a different approach for music. A way to work on heart rate, using the current activity which is in use and giving the user recommended music according to the heart rate of that of the user has been developed by the authors. It was certainly an unimaginable way for music recommendation which requires heart beat monitoring via smart band and mobile accelerometer which works in accordance with the android application. In this way leading your application towards sentiment analysis and with enough data providing the user with the best type of music in accordance to the model in an attempt to boost the mood of the user which is the essential part of user base.

Yajie Hu, Mitsunori Ogihara, [3] The authors have created a recommendation system for audio tracks and the research paper revolves around the implementation of that application. The writers say that it won't be reasonable to dump thousands of randomly related songs together without any sort of predefined albums or playlist. The analysis done by music streaming services takes something called forgetting curves into account which measure the freshness of the song and check for favoritism from the user's data while analysing what the user's listening patterns are. As per the authors, a good recommendation system is one that minimizes the efforts of user in providing needed feedback to owners and maximizing user

satisfaction. The recommendation system uses five basic factors: genre, year, favor, freshness, and time pattern. This will ensure successful implementation of NextOne players.

Anuja Arora, Aastha Kaul, Vatsala Mittal, [4] The authors of the following research paper talk about the influence of music on mood and emotional responses of the listener to it. They go on to discuss how it is able to detect emotions as well as lead to elevation of mood or in other words feeling better. Researchers segregated songs into 4 major moods and studied their effects and suggested songs based on the amount of data they have collected. They also take the user's facial expression into account and try to make a playlist for the emotion that the algorithm in itself detected. The used KNN, SVM and Haar cascades and also given us the accuracy for these models as well. The accuracy of the results were 70% for emotion detection accuracy and the various classes follower suitabilities were 81.6% and average followed by average music 92%. This gave us an application which plays music according to mood.

Xudong WU, [5] This paper's author has designed a music player of android operating system. The author implies that Android intelligent applications have fair share in the market. Thus, it is reasonable to create a music player which serves the higher userbase market for their own benefit. The paper gives us android OS architecture theory and makes a system structure that will help us create the media player. The writer giving a full detailed structure of app creation and all the add on features he will be adding are important part of. Writer divided the system into three major parts whose 1.) Main playback Interface 2.)Music Library Interface 3.) Download management interface. The features and functionalities of each of these three parts work in harmony with one another. Through this, the writer gives us an idea of how we can develop applications in the sense of media players in android operating systems.

Chapter 3

RESEARCH GAPS OF EXISTING METHODS

1. Generic Functionality Without Optimization:

- Popular apps have standard features but may not always implement them in the most **optimized or user-friendly** way.
- **Gap**: Missed opportunity to refine core features (like viewing story viewers) for better interaction and performance.

2. Limited Use of New Web Technologies in Educational Projects:

- While React.js and Firebase are common in industry, they are still underutilized in academic implementations of full-stack apps.
- **Gap**: A lack of real-world, full-stack implementations using **modern technologies** (like React, Firebase, and Node.js) in student-led academic projects.

3. Focus on Backend Logic Over User-Centric Design:

- Many existing projects tend to focus on technical backend implementation and neglect UI/UX aspects.
- **Gap**: Underrepresentation of **user experience enhancements**, particularly for common tasks like posting, browsing, or searching.

4. Lack of Component Reusability and Modular Design:

- Many implementations don't leverage the full capabilities of **React's reusable components**, affecting maintainability and scalability.
- **Gap**: Inadequate use of modular design practices for clean, reusable code and scalable front-end architecture.

5. Scattered Information Architecture in Social Media Clones:

- The user flow and data interactions (such as authentication, data retrieval) are often poorly documented or implemented.
- **Gap**: Need for a **clear, scalable, and secure architecture** combining real-time data handling and responsive design.

Chapter 4

PROPOSED MOTHODOLOGY

4.1 Functional Requirements

These define what the system should do:

- **User Requirement**:
  Users must create an account and understand basic app navigation to interact through posts, likes, and comments.
- **Device Requirement**:
  The application is platform-independent and runs smoothly on both Android and iOS operating systems.

4.1.1 Non Functional Requirements

These specify how the system performs under certain conditions:

- **Performance**:
  The app is expected to provide fast, accurate, and reliable responses.
- **Scalability**:
  The system should accommodate increasing numbers of users and data without performance degradation.
- **Availability**:
  The application should be accessible at any time as long as there is an internet connection.
- **Data Recovery**:
  In case of a server failure or error, mechanisms should prevent data loss wherever feasible.
- **Portability and Flexibility**:
  Users should be able to access the app from anywhere, anytime, using any compatible device.

4.2Development Tools and Technologies.

- Languages: JavaScript, HTML, CSS.
- Frameworks and Libraries (React.js, Node.js, Firebase)
- Visual studio code and npm.
- Use of Firebase authentication and firestore for authentication and database.

4.3 App Design Highlights

- **User Authentication**: Secure sign-up/sign-in using Firebase.
- **Post Creation**: Users can upload images with captions.
- **Story Functionality Enhancement**: A search bar within story viewers to improve navigation.
- **UI/UX**: Clean, responsive interface built with React and Material-UI.
- **Live Database Syncing**: Posts update in real-time using Firebase's snapshot listener.

Chapter 5

OBJECTIVES

5.1 Learning Objective

Learning objective of doing this project was first to get the understanding of how to build applications, the various API's in use and their functions, how to build a good UI and their applications to certain areas. Identifying the correct method to get to the solution we are trying to achieve while learning about how to use React js and HTML/CSS programming language into our application to give it the functionality required for the proper working of an app.

5.2 Outcome Objective

The main outcome objective was to create a framework with similar functionalities to that of a social media application which works on any device.The implementation and coding has been done on windows 10. Then using technologies like Google's Firebase and React Js implement the things like User authentication, Database Management to develop a dummy social media application. We can create an account and post images with captions and comment on the image. The process is made simpler by using API's and their inbuilt function calls which predefine some important features needed for the app.

Chapter 6

SYSTEM DESIGN & IMPLEMENTATION

Software Used :-

Visual studio, Npm

Language Used :-

Java Script, HTML, CSS

Main Library :-

React.js, Node.js

6.1 About Visual studio

The code editor which has been designed and developed by tech giant Microsoft for operating system platforms like Windows10, MacOs and Linux is VS Code. The application possesses debugging features that include syntax highlighting, autocompletion and code hinting. It also includes code-snippets, refactoring features and Git VCS for version control. Such features back a good IDE to help us write better code to get the task done efficiently. Visual Studio features a code editor that comes with a feature called IntelliSense that automatically completes codes. Also, it does code restructuring as a process. The debugger that is installed can work as a machine-level and a source-level debugger. Tools for designing classes and database schemas are also included. This tool may install or uninstall plug-ins that enhance technical functionality at all levels. This means that it would install editors for other software development life cycle stages as well as to input support for the source control systems. A mixture of data and knowledge is usually the new age word that carefully creates the 21st-century technology. It is 27 years old and it was first published in 1995. Artificial intelligence and data science are the most demanding professions nowadays. The Community edition, is available for free.

6.2 About npm

The full form of npm is node packet manager. It is basically used for JavaScript which is a programming language used by html. NPM is a GitHub's product or can we say GitHub's subsidiary which works as a server actually a host for developing software and version control of software using Git and Version Control System is a git. Node packet manager

default package. JavaScript programming language manager. According to an Oxford University study, the use of open-source software has rapidly risen since 1991. The escalation usage is due to its power and capability. The node packet is something that is making this possible for SIG and JavaScript. Yet, several organizations utilize it for proprietary uses that are not publicly available.

Npm is made up of 3 parts.

- website.
- registry.
- Command Line Interface (CLI).

We use npm to set up the profile and use it for discovery of packages and many other things. For instance, you can find organizations to administer the grant to packets. Private or public packets. Most developers interact with npm using functions in terminal like cmd or git which is CLI. The npm registry, which is a large database, contains JavaScript and its meta information.

6.3 About React

React is a framework that speeds up builds and makes the pain of building interactive user interfaces easy. When using React, we can create views for any stage of your application development without much effort. React is very efficient with its updates, only changing the components that need to change rather than the whole application. Through the use of declarative views, code interpretations become easier while debugging and optimizing. Stateful components are elements that have states of their own and can be put together to create nicer UIs. In JavaScript, the logic that allows these components to function is better than a template as it enables ease of data passing in our app. We do not assume any technology that will be used by default, furthermore, we will be able to add many new features in React without re-writting the existing app. Node also enables application logic on the server side in addition to ReactJs. Whereas with React Native, powerful apps can be developed. To gain a deeper understanding of React, let’s discuss how it actually works in the background i.e. background processes. ReactJs is one of the highest performing scripting languages employed today. Additionally, React operates quickly and can run on very low memory. Because React abstracts the DOM (Document Object Model), there is a separation of concerns that is taking place. Most front-end technologies for data binding implement either Key-Value Observation or Object Observation. Examples include Meteor, Ember, Angular, React and Vue.js. Dirty

Checking like (AngularJS). React moves differently and functions on a. javascript approach. We will see how React works in its implementation: First the DOM Abstraction happens and a virtual representation of the DOM is created that is stored in the. memory. The React component lifecycle begins when a component is created . This means that the component is rendered for the first time. After that, the created component will not vanish, but it will be unmounted when it is not rendered. When created and mounted on the DOM, the component goes through a series of events that takes it to the next stage.

The mounting is when the components are being built from scratch, which means they are created for the first time. The next stage will be when the component is being modified, and they are already present in the latest version of the DOM.

6.4 About Node.js

Node.js was designed like other frameworks Ruby and Twisted python with capabilities to support event machines. Node.js implements an event model, which is used in a number of implementations. The runtime construct of node.js is event loop, not a library. In many other systems there will always be a blocking call which starts the event-loop. Typically, behavior is defined through callbacks that occur at the beginning of any script, and at the end, a server is started through a blocking call. There is no call to start-the-event-loop. Node.js executes the input script before it enters into the event loop. When there are no more callbacks remaining, Node.js terminates the loop. It behaves similarly to the JavaScript of a browser. The event loop is not exposed to the user. Node.js' main component is HTTP which helps with low latency and streaming support. (16 words) Node.js is a good choice for building a library framework. If you don’t design your Node.js programs with threads in mind, you lose out on the multiple cores that you have in your environment. We can use API child_process.fork() to create child processes that are easy to communicate with and are based on the same interface as a cluster module that permits sharing of sockets between processes for load balancing over multiple cores.

6.5 Flowchart

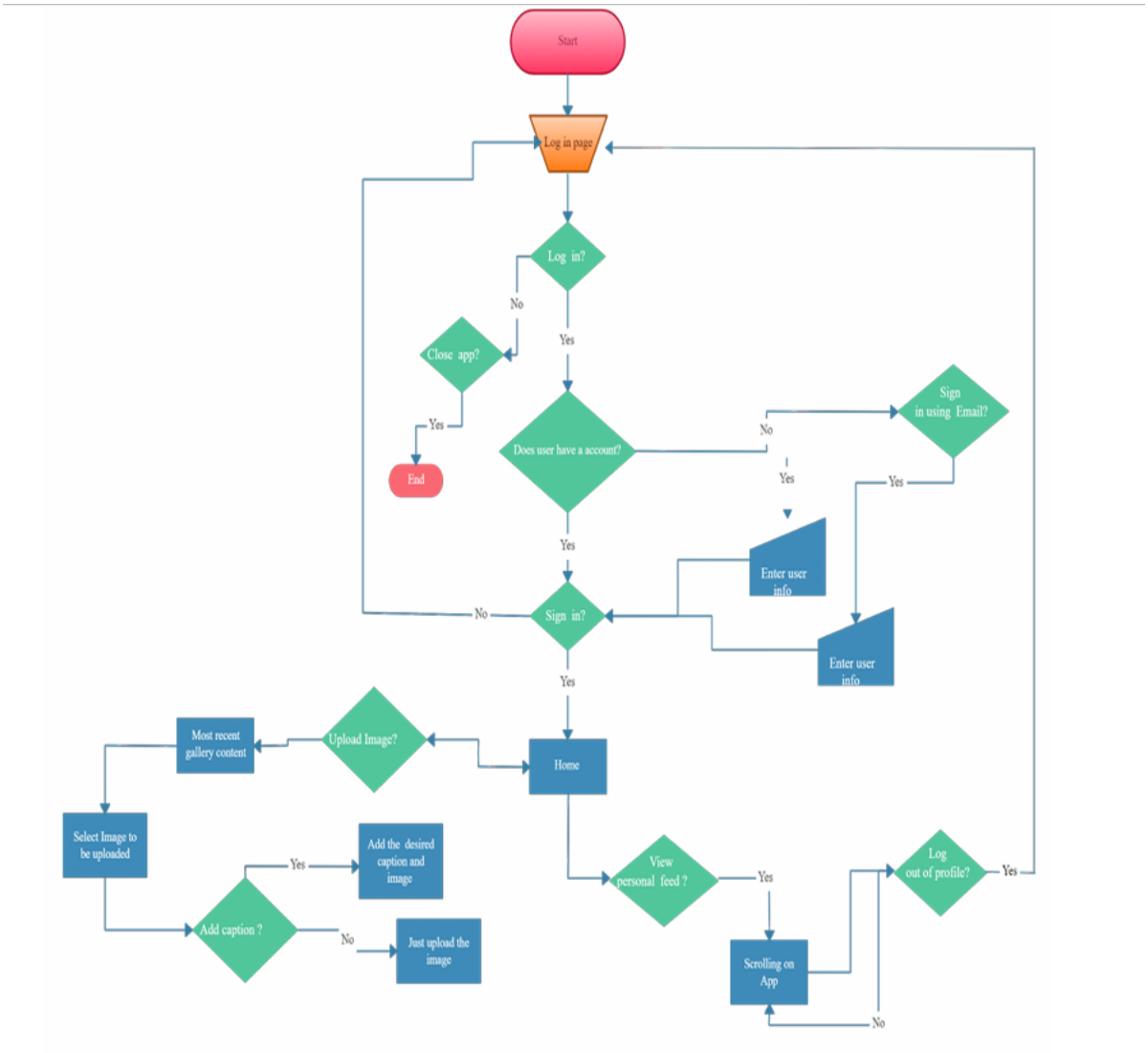


Fig 1.1. Flow diagram depicting procedures to be followed in app

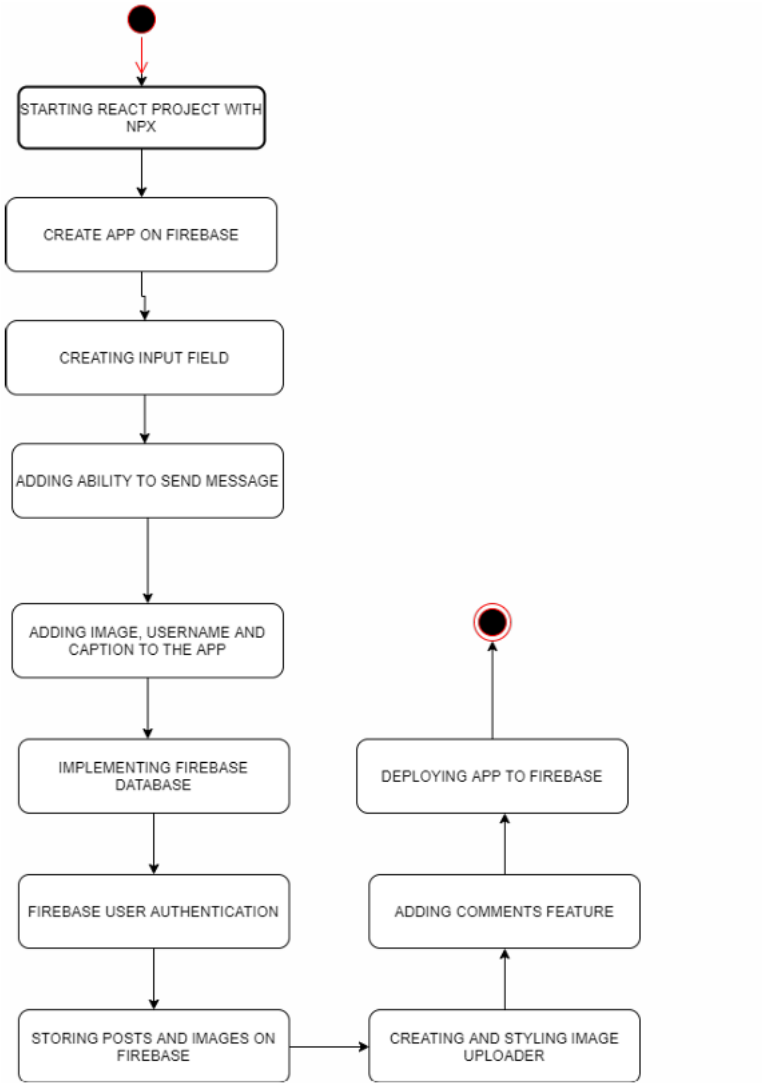


Fig 1.2 IMPLEMENTATION OF PHASES/STAGES (DIAGRAM)

Chapter-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

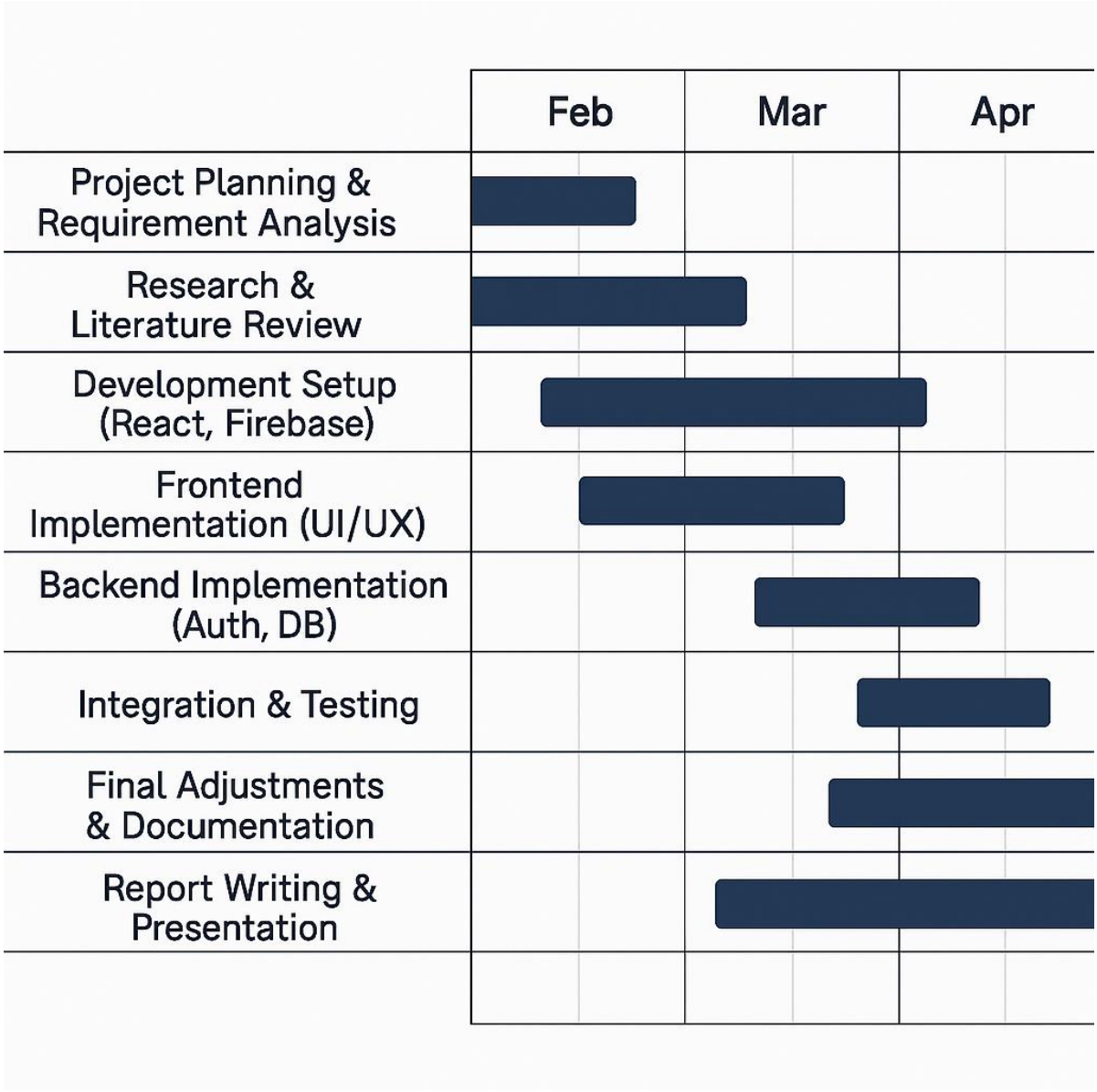


Fig 1.3 Gantt chart

Chapter 8

OUTCOMES

8.1 Functional Prototype Delivered

- Developed a **fully functional web-based social media application**.
- Core features included:
    - User **authentication** via Firebase
    - **Image uploads** with captions
    - A real-time **feed** display
    - **Comments section** for interaction

8.2 User Interface & Experience

- Clean and responsive UI designed using **React JS** and **Material UI**
- Smooth navigation flow, aligned with **popular social media platforms**
- Integrated **search functionality** for better story interaction

8.3 Firebase Integration

- **Firebase Authentication**: Handles sign-up and login
- **Cloud Firestore**: Real-time data storage and syncing
- **Firebase Storage**: Handles user image uploads securely

8.4 Tools & Tech Stack Mastery

Hands-on experience with:

- React JS, Node.js
- npm, Visual Studio Code
- Firebase suite (Auth, Firestore, Storage)

Chapter 9

RESULTS AND DISCUSSIONS

9.1. RESULTS

9.1.1. Implementation Results

Through the development process, the team successfully implemented a social media web application using React JS with the following working features:

User Authentication:

Implemented using Firebase. Users can sign up, sign in, and sign out securely.

Posting System:

Authenticated users can upload images, add captions, and share posts.

Real-Time Updates:

The app dynamically displays new posts and updates using Firebase's Firestore real-time database.

User Interface:

The UI closely mimics popular social media platforms. React components and Material-UI were used to ensure responsiveness and modularity.

Image Upload:

Users can upload images directly from their devices and store them in Firebase Storage.

Feed Display:

Posts are ordered by timestamp, providing a timeline-like experience

9.1.2. Performance Highlights

Speed:

React's virtual DOM and Firebase's fast query execution provided quick rendering and real-time updates.

Scalability:

The app structure and Firebase's backend allow it to support many users with minimal changes.

Responsiveness:

The app was tested on various screen sizes and maintained functional consistency.

Security:

Firebase Authentication ensures secure login and protects user data from unauthorized access.

Table 1- Analysis of clone detection techniques

Techniques	Approaches used	Parameters for detection	Implemented on	Advanced Features	Outcomes
Content Based.	Attribute Similarity.	First Name, Surname, Education, Company, City, Country.	Facebook, LinkedIn, MeinVZ, StudiVZ, XING.	Offer Automated crawling and CAPTCHA.	Friend request acceptance of profile cloning: over 60% and cross site profile cloning: 56.4%.
		Surname, current and past employment, education, Image comparison.	LinkedIn.	Image comparison.	100% detection of clone profiles.
	Attribute similarity and Friend Network Similarity.	Friend list, Recommended friend list, Excluded friend list.	Facebook.	Provide detection on Basic Profile Similarity (BPS) and Multiple-Faked Identities Profile Similarity (MFIPS).	114 more faked identities were detected in MFIPS approach than BPS approach.
	Profile Analysis for a Time Interval.	Number of friends, real social interaction, structure of online social network graph.	Facebook.	First one to analyse social network graphs from dynamic point of view.	Not able to fully reconstruct day-by-day graph of the users network starting from 1st day.
		User icon, background image, screen name, number of followers and friends, status updates (number of updates, sources, frequency and contents).	Twitter.	Pearson's χ^2 test on number of tweets: minutes-in-the hour and seconds-in-the-minute.	Detected 16% accounts which publish tweets that exhibit discernible automation.
Content Free.	Attribute similarity and Record of IP address.	Name, gender, school, living city, first four recent 16bits of IP address.	Renren.	Snowball Sampling and Iteration attack.	Suffers from IP Spoofing.

Table 9.1 Analysis of clone detection techniques

9.2. Discussion

UI/UX Optimization:

The inclusion of a search bar within the story view significantly enhanced usability—especially for users with large follower counts.

Component-Based Approach:

Using React's modular structure allowed for faster development, debugging, and future scalability.

Challenges Faced:

Handling Firebase a sync functions and state updates required careful logic structuring.

Image upload limits and browser-based testing had constraints on performance under heavy data loads.

Learnings:

Mastery over React hooks like useState, useEffect

Firebase integration from scratch (Firestore, Auth, Storage)

UX thinking from a user-first perspective

Chapter 10

CONCLUSION

In this short article, we have given a brief highlight of building social media app using reactjs. Generally react.js allow us to design reusable user interface components. It also allows developers to create large web applications which help in changing the data without reloading the page. People use React because it's easy to learn, scalable, fast, and simple, and we made our app using it for its feature like accurate, quick, scalable, and simple. In reacts we have used npm and node.js.React is basically an excellent tool with which we can create some interactive applications for mobiles,web and any other platform.Some other advantages of using Reacts are:

- Having better efficiency and overall developer productivity
- You can share your code and can reuse
- High performance with some advance features
- It provides large number of free of tools

React introduced us to the JavaScript which is used for web development.JavaScript is a test-based programming language that used on both client and server side which help us to make our application more interactive with its advanced features whereas HTML and CSS are language that generally use for give the structure and style to application Javascript gives interactive element.We conclude our project by telling you that social media app using Reacts has been implemented.

REFERENCES

[1] : Junli Xu :" Design and implementation of the music player" :Conference: 2015 International Conference on Management, Education, Information and Control (2015)

[2] : Ravi Kumar, Numer Mulla, Dishank Gangar :"Illusion: A smart music player": International Journal of Engineering Research and Technology (2020)

[3] :Yajie Hu, Mitsunori Ogihara :"Nextone player: A music recommendation system based on user behaviour":(2011)

[4] :Anuja Arora, Aastha Kaul, Vatsala Mittal :"Mood based music player":2019 International Conference on Signal Processing and Communication (ICSC)":(2019)

[5] :Xudong Wu :"Design of Music Player Software Based on Android ":6th International Conference on Machinery, Materials and Computing Technology: (2018)

APPENDIX-A

PSUEDOCODE

Index.html

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link
      href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;500;600;700;800&display=sw
      rel="stylesheet"
    />
    <style>
      * {
        font-family: "Open Sans", sans-serif;
        margin: 0;
        padding: 0;
      }
    </style>
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>

```

Index.js

```js
import express from "express";
const app = express();
import authRoutes from "./routes/auth.js";
import userRoutes from "./routes/users.js";
import postRoutes from "./routes/posts.js";
import commentRoutes from "./routes/comments.js";
import likeRoutes from "./routes/likes.js";
import relationshipRoutes from "./routes/relationships.js";
import cors from "cors";
import multer from "multer";
import cookieParser from "cookie-parser";

//middlewares
app.use((req, res, next) => {
  res.header("Access-Control-Allow-Credentials", true);
  next();
});
app.use(express.json());
app.use(
  cors({
    origin: "http://localhost:3000",
  })
);
app.use(cookieParser());

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "../client/public/upload");
  },
```

```
  filename: function (req, file, cb) {
    cb(null, Date.now() + file.originalname);
  },
});

const upload = multer({ storage: storage });

app.post("/api/upload", upload.single("file"), (req, res) => {
  const file = req.file;
  res.status(200).json(file.filename);
});

app.use("/api/auth", authRoutes);
app.use("/api/users", userRoutes);
app.use("/api/posts", postRoutes);
app.use("/api/comments", commentRoutes);
app.use("/api/likes", likeRoutes);
app.use("/api/relationships", relationshipRoutes);

app.listen(8800, () => {
  console.log("API working!");
});
```

Profile.js

```
import "./profile.scss";
import FacebookTwoToneIcon from "@mui/icons-material/FacebookTwoTone";
import LinkedInIcon from "@mui/icons-material/LinkedIn";
import InstagramIcon from "@mui/icons-material/Instagram";
import PinterestIcon from "@mui/icons-material/Pinterest";
import TwitterIcon from "@mui/icons-material/Twitter";
import PlaceIcon from "@mui/icons-material/Place";
import LanguageIcon from "@mui/icons-material/Language";
import EmailOutlinedIcon from "@mui/icons-material/EmailOutlined";
import MoreVertIcon from "@mui/icons-material/MoreVert";
import Posts from "../../components/posts/Posts";
import { useQuery, useQueryClient, useMutation } from "@tanstack/react-query";
import { makeRequest } from "../../axios";
import { useLocation } from "react-router-dom";
import { useContext } from "react";
import { AuthContext } from "../../context/authContext";
import Update from "../../components/update/Update";
import { useState } from "react";

const Profile = () => {
  const [openUpdate, setOpenUpdate] = useState(false);
  const { currentUser } = useContext(AuthContext);

  const userId = parseInt(useLocation().pathname.split("/")[2]);

  const { isLoading, error, data } = useQuery(["user"], () =>
    makeRequest.get("/users/find/" + userId).then((res) => {
      return res.data;
    })
```

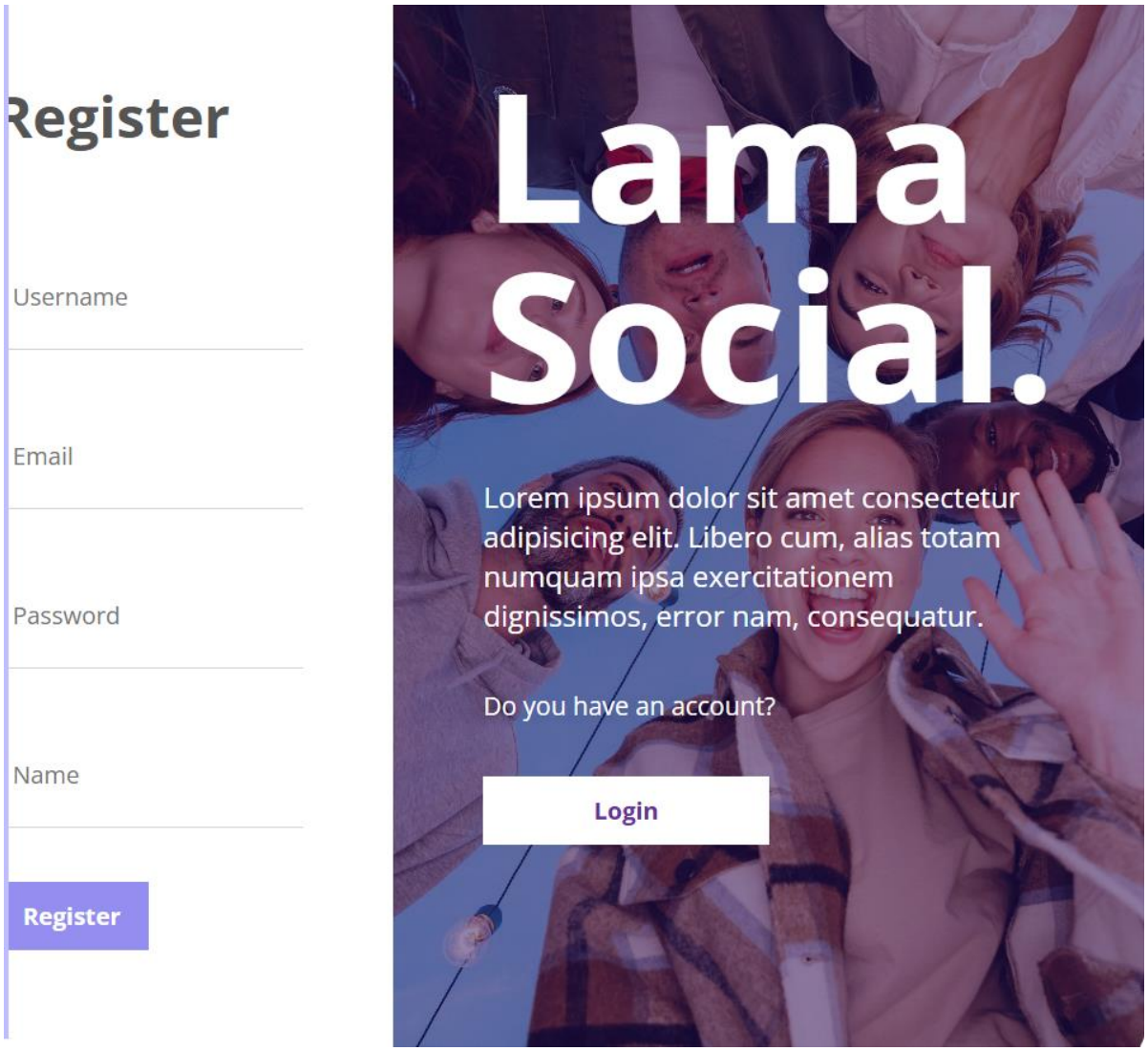
CSS Code:

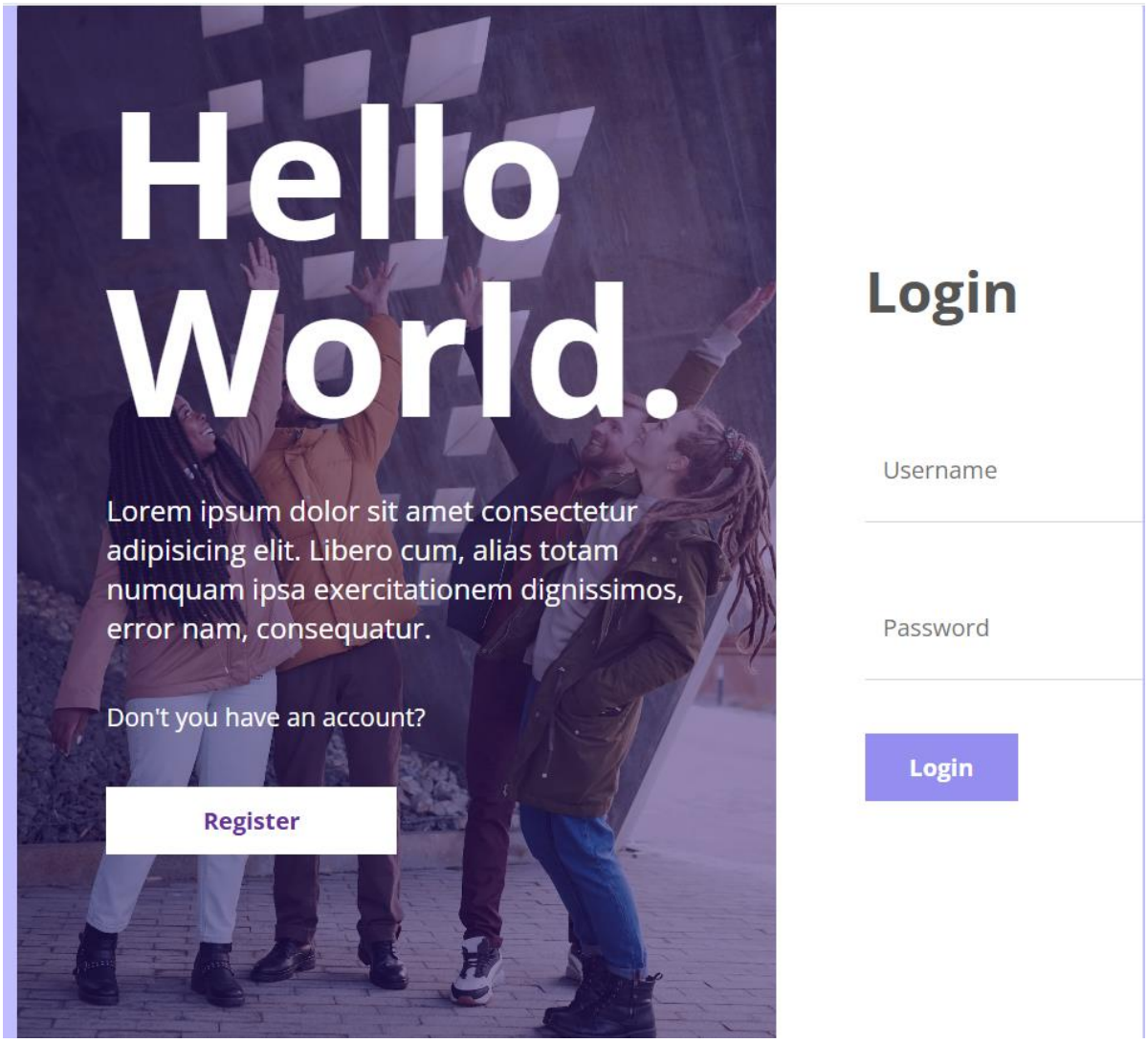
```
* {
margin: 0;
}
.app {
background-color:  #fafafa;
height: 100vh;
}
.app_headerImage {
background-color: white;
padding: 20px;
border-bottom: 1px solid  lightgray;
object-fit: contain;
}
post {
background-color: white;
max-width: 500px;
border: 1px solid lightgray;
margin-bottom: 45px;
}
.post_image {
width: 100%;.
object-fit: contain;.
border-top: 1px solid lightgray;.
border-bottom: 1px solid lightgray;.
}.
Make the font-weight of your text normal.
Give a new version of this.
.post_avatar {.
margin-right: 10px;.
}.
```

APPENDIX-B

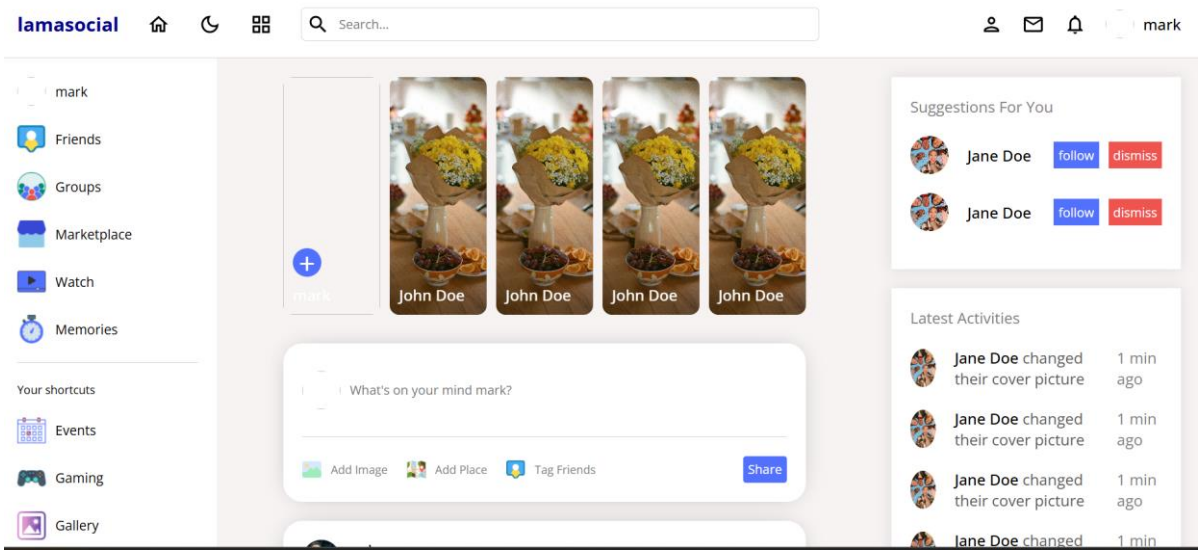
SCREENSHOTS

Login Page :-

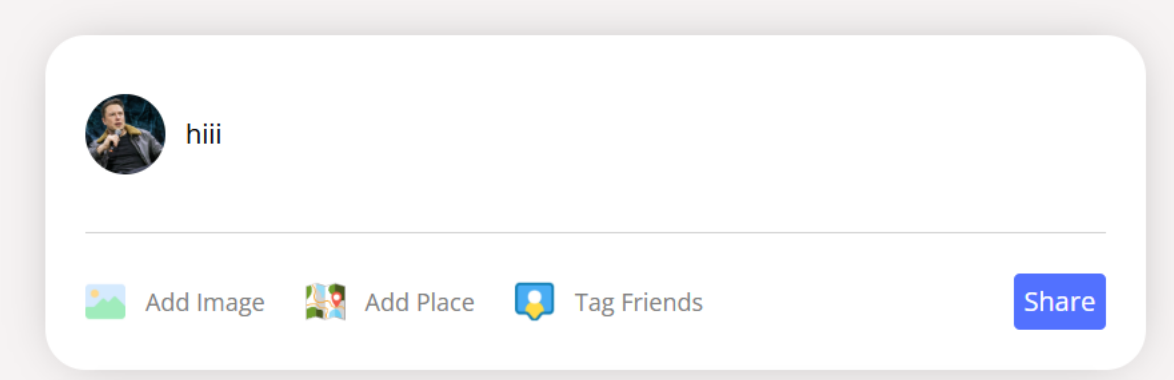


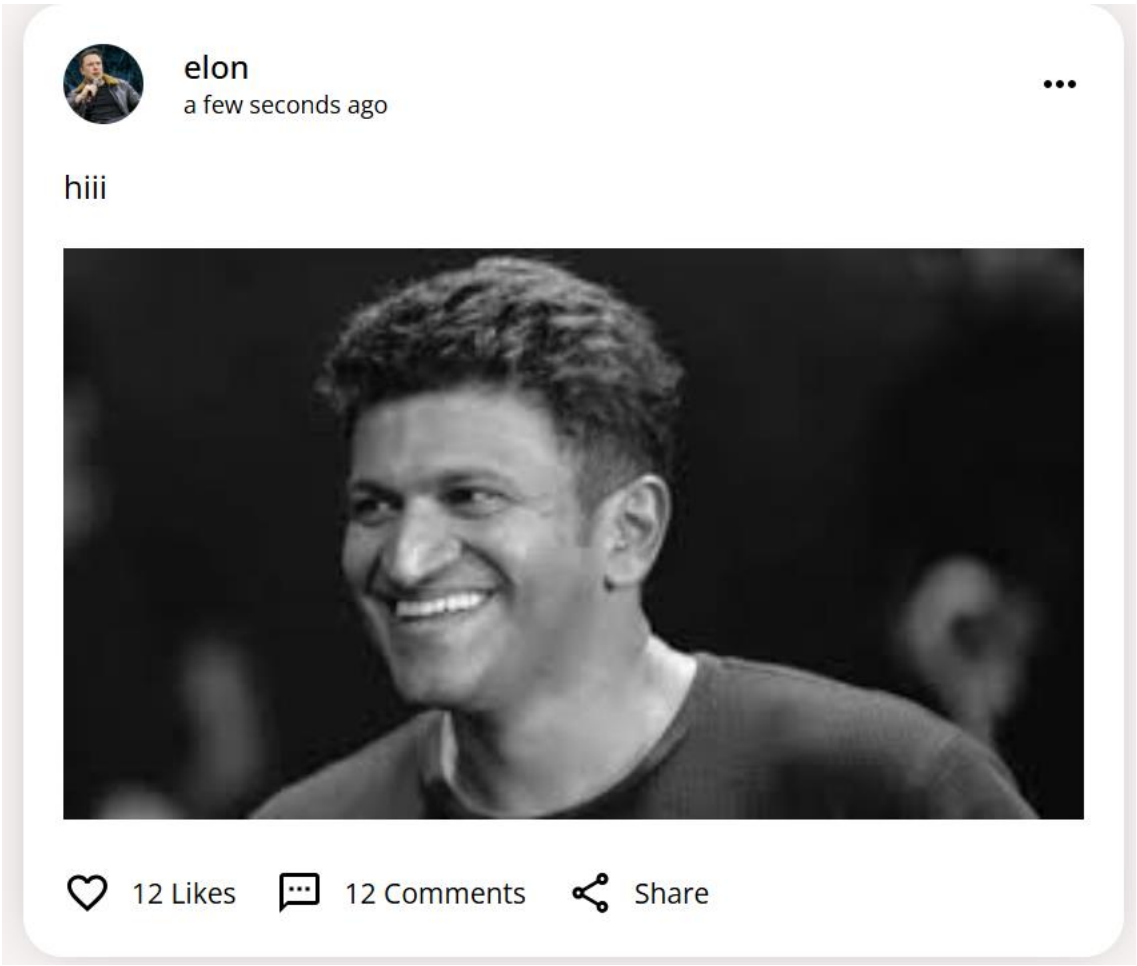

Hello World.
Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero cum, alias totam numquam ipsa exercitationem dignissimos, error nam, consequatur.
Don't you have an account?
Register
Login
Username
Password
Login

Home Page :-

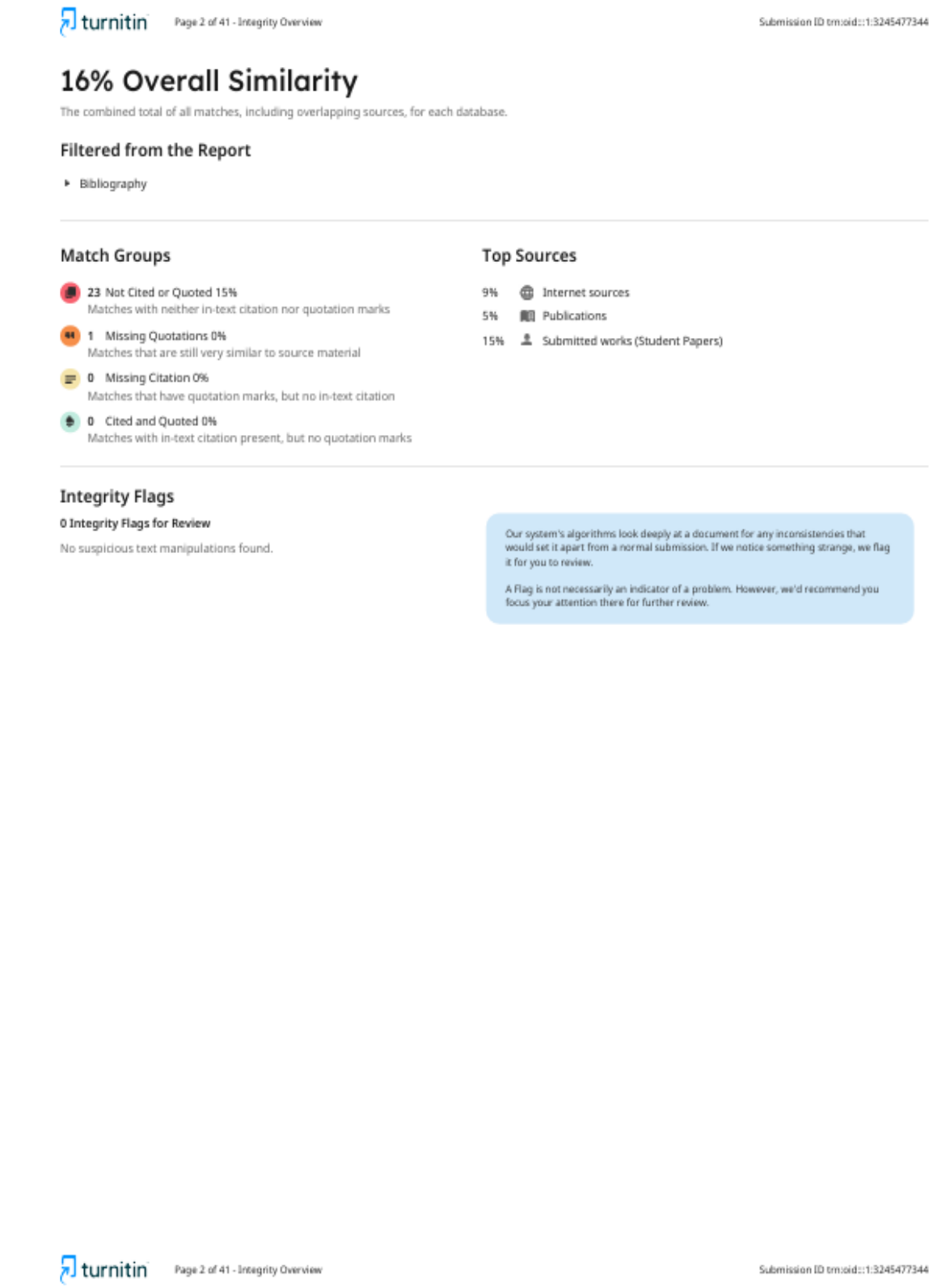


Post :-




elon
a few seconds ago
hiii
12 Likes
12 Comments
Share

APPENDIX-C



turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::1:3245477344

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

▸ Bibliography

Match Groups

- 23 Not Cited or Quoted 15%
  Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
5% Publications
15% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 41 - Integrity Overview Submission ID trn:oid:::1:3245477344

turnitin Page 3 of 41 - Integrity Overview Submission ID trn:oid:::1:3245477344

Match Groups

- 23 Not Cited or Quoted 15%
  Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
5% Publications
15% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
Presidency University		11%
2	Student papers	
M S Ramaiah University of Applied Sciences		1%
3	Student papers	
Manipal University		1%
4	Student papers	
Lovely Professional University		<1%
5	Internet	
ebin.pub		<1%
6	Internet	
research.library.mun.ca		<1%
7	Student papers	
Tata Institute of Social Sciences		<1%
8	Internet	
etd.aau.edu.et		<1%
9	Student papers	
Symbiosis International University		<1%
10	Internet	
edyaaleh.files.wordpress.com		<1%

turnitin Page 3 of 41 - Integrity Overview Submission ID trn:oid:::1:3245477344

turnitin Page 2 of 40 - AI Writing Overview Submission ID trn:oid:::1:3245477344

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

turnitin Page 2 of 40 - AI Writing Overview Submission ID trn:oid:::1:3245477344

SUSTAINABLE DEVELOPMENT GOALS

The project work carried out here is mapped to the below 2 goals:

- **Goal 16 – Peace, Justice, and Strong Institutions**
- **Goal 17 – Partnerships for the Goals**