

SOCIAL MEDIA APP USING REACT JS

Project report submitted in partial fulfillment of the requirement for the degree of
Bachelor Of Technology

In

Computer Science and Engineering/ Information Technology

By

Falaq Shahraj Khan(171354)

Shubhra Pundir(171262)

Under the supervision of

Dr. Kapil Sharma

To



Department of Computer science & Engineering and Information Technology

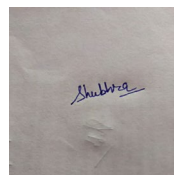
Jaypee University Of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

We hereby declare that the work presented in this report entitled “ **SOCIAL MEDIA APP USING REACT JS**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2021 to May 2021 under the supervision of **Dr. Kapil Sharma** (Assistant Professor(SG), Computer Science and Engineering And Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Falaq Shahraj Khan(171354)



Shubhra Pundir(171262)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Supervisor Name: Dr. Kapil Sharma

Designation: Assistant Professor (SG)

Department name: Computer Science and Engineering And Information Technology

Dated:

ACKNOWLEDGEMENT

Researching and working on our project “Social media app using react js ” has been a great learning experience. It introduced us to the vast world of APIs, UI/UX development, and their use in building applications for ease of our day-to-day lives. We got to learn about how to build an application in the right manner while learning about how to do programming with ReactJs and components like material UI.

We would like to express our special thanks and gratitude to ourselves for completing this project and working on it till the end while being in a pandemic(Covid) and not giving up, and also to our mentor **Dr. Kapil Sharma**.

Falaq Shahraj Khan(171354)

Shubhra Pundir(171262)

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.

TABLE OF CONTENTS

I.	Certificate	01-02
II.	Acknowledgment	03
III.	Abstract	04
IV.	List of Abbreviations	05
V.	List of Figure	06
VI.	List of Graph	07
VII.	List of Table	08
Chapter 1	Introduction.....	10-14
1.1	Introduction.....	10
1.2	Problem Statement.....	11
1.3	Objective	12
1.4	Methodologies.....	13
1.5	Organization.....	14
Chapter 2	Literature Review.....	15-18
Chapter 3	System Development.....	19-24
3.1	About Visual Studio.....	19-20
3.2	About npm.....	2-22
3.3	About React.....	22-23
3.4	About Node.js.....	21-22
3.5	Flowchart.....	22
3.6	IMPLEMENTATION OF PHASES/STAGES (DIAGRAM).....	23
Chapter 4	Performance Analysis.....	24-38
Chapter 5	Conclusions.....	39
References.....		40

LIST OF ABBREVIATIONS

IDE -Integrated Development Environment

API -Application Programming Interface

UI-User Interface

E-ADT-Eclipse Android development tools

GUI- Graphic User Interface

VCS- Version Control System

LIST OF FIGURES

Figure no.	Description	Page No.
3.1	FLOW DIAGRAM DEPICTING PROCEDURES TO BE FOLLOWED	22
3.3	IMPLEMENTATION OF PHASES/STAGES (DIAGRAM)	23
4.1	UI OF SOCIAL MEDIA APP	36
4.2	IMAGE UPLOADER	37
4.3	UI	37
4.4	FINALIZED LOOK OF APP	38

LIST OF GRAPHS

Graph no.	Description	Page no.
3.1	Flow diagram depicting procedures to be followed in app	22

LIST OF TABLES

TABLE NO.	DESCRIPTION	PAGE NO.
01	Analysis of clone detection technique	24

CHAPTER 1

INTRODUCTION

1.1 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 be developed on new technologies like React.js and Material UI while using Google's Firebase for Authentication and Database management. The Web app lets you post images on your feed or story for a specific period of time after which the story disappears. You can view who has seen your story via checking the usernames that are provided in the story section. It is easy to navigate through the list if the list is short but imagine doing so with 10K followers or even 200 followers. The process of scrolling to see who has viewed your story gets hectic and a lot time-consuming.

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. Simple:- With React, you express how your app should look at any given point in time - and it will automatically manage your UI updates when your underlying data changes.

2. Declarative:- With changes in Data, React automatically refreshes with it and it knows to only update the changed parts and not the whole UI. This improves the overall performance.

3. Reusable:- With React we can build reusable components, we majorly build components with the help of react. As they are encapsulated components it makes the code reuse, testing, and removal of errors pretty easy.

Simple, declarative, and reusable. These are the top 3 major qualities of React.

1.2 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 ReactJs and 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.

1.3 Objectives

1.3.1 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.

1.3.2 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.

1.4 Methodology

1.4.1 Functional Requirement

The functional requirement specification of the project are mainly categorized as user requirements, security requirements, and device requirement each of which are explained in detail below:

- i. **User Requirement:** User ought to have an account on framework and client must know how to interact with the system so that he can further post , like or comment on the framework.
- ii. **Device Requirement:** User can use an android or ios operating system and the app will run fine on both the platforms.

1.4.2 Non-functional Requirement

- i. **Performance:** The framework will have speedy, exact and dependable outcomes.
- ii. **Capacity and Scalability:** The framework will have the capacity to store identity registered by the framework into the database.
- iii. **Availability:** The framework will be accessible to clients whenever at whatever point there is an Internet association.
- iv. **Recovery:** If a break down or a server is inaccessible, the application should have the capacity to keep any information or data loss from happening in the optimal conditions.
- v. **Flexibility and Portability:** System will be available whenever from anywhere.

1.5 Organization

In Chapter 1, we have discussed what is social media app and it's

development features while being in development mode. Also, we pointed out the schemes that we can use in our project report.

In Chapter 2, we will be discussing the research papers we have referred to in order to get a better understanding of our project. The papers mainly focus on techniques used in front end development, android app development and other researches carried out in this field.

In Chapter 3, we will be citing the possible requirements that are the hardware and software system, what language we will be using and where we are going to implement it along with the libraries required along with details about the platform used.

In Chapter 4, we will be discussing in detail the algorithms and approaches used to predict the outcome and effectiveness of our result. Implementations and the results of the outputs have been discussed.

In Chapter 5, we will be giving the conclusions that have been derived from this study and the future scope of this project.

CHAPTER-2

LITERATURE SURVEY

2.1 Literature Review

Junli Xu, [1] The author has classified android as an open source operating system unlike previous closed source operating systems and says that with an open source OS it certainly gives us the opportunity to work on whatever we want for ourselves. After this the author goes precise and starts to talk about the music player. The author says that as much of pretty UI is favoured by people it surely leads the mobile device towards more resource utilization. Author says that he is making a media player for android which has certainly less resource utilization while being simple, convenient and user-friendly as well. He uses JAVA as their main language and eclipse as the IDE for implementing this approach. The author in this manner develops an android application with playlists, main interface, menus and play settings.

Ravi Kumar, Numer Mulla, Dishank Gangar, [2] 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. So the authors have decided to create a smart music

application named Illusion which takes into account clearly a different approach for listening to music. The authors have developed a method to work with the heart rate data, the ongoing activity that is currently in action and providing the user then with the recommended music in accordance to the heart rate of that of the user. 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 while trying to boost the mood of the user who is an integral part of the 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 authors point out that it would not be practical to store thousands of unrelated songs together without some set of predefined albums or playlists. They take something called a forgetting curve into account which measures the freshness of the song and check for favoritism from the user's data while analysing what the user's listening patterns are. The authors also point out that a good recommendation system minimizes the user's effort required to provide the owners with necessary feedback and maximize the user's satisfaction. The recommendation system is built on 5 fundamentals like genre, year, favor, freshness and time pattern. Thus leading towards a successful implementation of NextOne players.

Anuja Arora, Aastha Kaul, Vatsala Mittal, [4] The authors of the following research paper talk about how mood is influenced by music and provides the listener with an emotional response to it. They further talk about how it can detect emotions as well as lead towards elevation of mood or in other words feeling better. They have classified the songs into four major mood categories and studied the effects while trying to suggest songs based on the amount of data they 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. They have used models like KNN, SVM and Haar cascades and also provided us with the accuracy for these models. The respective accuracies were 70%, 81.6% and 92% and gave us an application which plays music according to mood.

Xudong WU, [5] The author of this paper has developed a music player for the android operating system. He signifies the importance of android intelligent applications and it's fair share in the market concluding that it is rather natural to build a music player which server the higher userbase market for personal gains. The paper provides us with the architecture theory for android OS and developing a system structure for the media player. The author provides a full detailed structure of how the app creation takes place and what are the features he will be adding are an integral part of. The author has divided the system into three major parts which are 1.) Main playback Interface 2.)Music Library Interface 3.) Download management interface. All three of these parts have their own features and functionalities that work hand in hand. In this manner the author gives a brief overview of app

development in context of media players working on android operating systems.

D. Muriyatmoko, N A S Asy'ari, M S Arif [6] The authors of this research paper have developed an android application which is used for streaming radio from Songgolangit FM Ponorogo. They stated that the broadcast signal was certainly limited in Ponorogo. The method used in the android application development was the waterfall model and they have performed black box testing on this model. The idea was to improve the listener base coverage for Songgolangit FM. They have also provided us with data about their system development approach and how they used it in order for their app to work. They have created a basic app with necessary functionalities for the radio like description of the radio, play and pause buttons and the Songgolangit radio website divert button.

CHAPTER 3

SYSTEM DEVELOPMENT

Tools and technologies used:

Software Used:-

Visual studio, Npm

Language Used:-

Java Script, HTML, CSS

Main Library:-

React.js,Node.js

3.1About Visual studio

The Visual Studio Code is a free open source code editor which is made by the tech giant Microsoft for operating system platforms like Windows10, MacOS and Linux.It has features that will support the application in debugging of the code , highlighting the syntax and auto completion of the code by giving hints as to what we might be writing, code-snippets, refactoring of code, and by using git VCS managing version . Visual Studio has a code editor which supports a feature called as IntelliSense (the component which completes the code) and it also does code refactoring as a process. The debugger which has been installed works like a source-level debugger as well as a machine-level debugger. Other tools which are built in

include class designer and database schema designer. It can install or uninstall plug-ins that improve the technical functionality at all levels which include inputting support for control systems of source and installing editors for other stages of the software development life cycle.

Microsoft Visual Studio comes with support for development in 36 different programming languages and it grants the code editor and debugger support to nearly any programming language. The Community edition, is available for free.

3.2 About npm

npm stands for node packet manager. It functions as a package manager for programming language mainly speaking -JavaScript. npm is a product of GitHub or we can say it's github subsidiary, which gives a host like service i.e server for development of software and control of Version by using Git as a version control system. node packet manager is the default package manage of the programming language which is JavaScript. Interestingly enough, node packet manager is the world's largest software registry. The developers which contribute to open source from every place in this world use node packet manager to give and take data in form of packets, however many organizations use it for private uses also which are not available to the general population.

npm has 3 major parts :

- website
- registry
- Command Line Interface (CLI)

We use npm for package discovery, setting up a profile, and managing various other things of npm usage. Let's clear it with an example, you can identify organizations to manage the grant to public or private packages. The CLI functions from the terminal generally cmd or git, and is how most developers interact with npm. The npm registry acts as large open-source database of JavaScript and it's meta information.

3.3 About React

React is a framework which makes creating interactive UI's a lot less time consuming and makes it painless to create interactive UIs. React lets us design easy to create views for every stage in our application development, React has a great efficient update and it works on the components that need the change not on the whole application. The views which are declarative in nature makes the code easy to analyze and even easier to optimize and debug it. Building components which are encapsulated and manage their own state, we combine them to make better UI's. The logic of the components is written in JavaScript language and not as templates, we can pass data with ease through our app. No assumptions are made about what technologies a person might be using, we can develop new features in React without having to again rewrite the existing code. ReactJs can also make changes on the server with the help of Node and can make powerful apps via React Native. To understand react in more depth, let us discuss the functioning of React in the background i.e background processes. The best and most important benefit of ReactJs is performance. The speed of React is a thing of beauty, and it works on low memory principles that are achieved by abstraction of the DOM (Document Object Model) with a virtual DOM in action. To implement data binding most of the front-end technologies use one of the two i.e Key-Value Observation like (Meteor, Ember) or the Dirty Checking like (AngularJS). React acts in a different manner and acts on a Javascript approach. Let's see how React works on this implementation:

First happens the DOM Abstraction and a virtual representation of the DOM that is stored in the memory.

Now when this happens, the data model shifts/changes, React then goes through the process of re-rendering and only the components that use the data that has changed "Diffs" the previous version of virtual DOM with the new version of virtual DOM updates to the actual DOM, and leads to only modifying those components which require this change and will be directly beneficial to them.

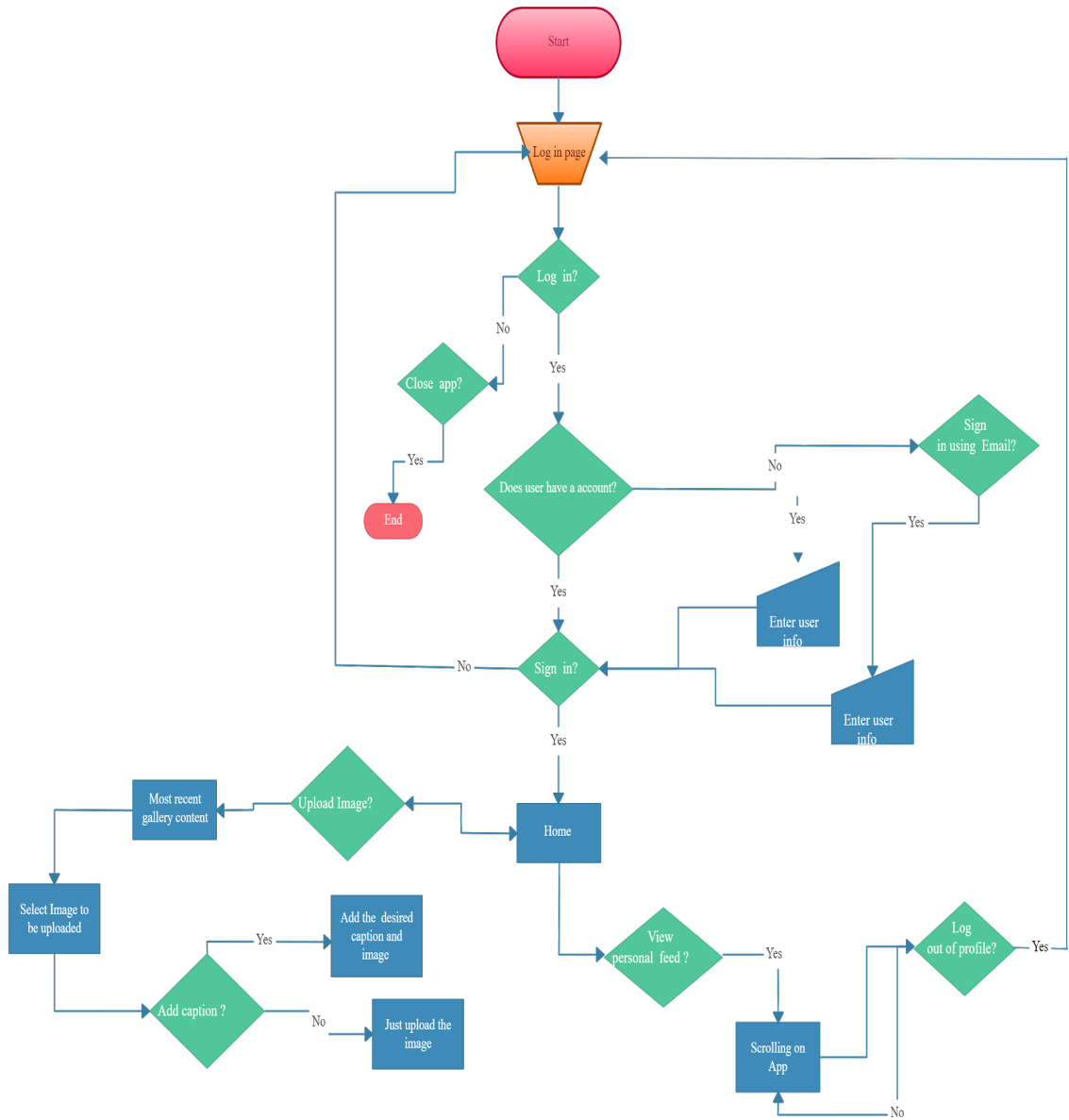
3.4 About Node.js

Node.js has a somewhat similar design, and is generally in influence of, systems which are for example like the event machine of Ruby and Twisted-python. Node.js functionality picks up the event model and takes it to a different level. Node.js gives event loop like a runtime construct and not like a library. In many other systems, a blocking call to start the event-loop will always be there. The behaviour is generally defined via callbacks which are done at the beginning of any script, and a server is started via a blocking call

at the end. There is no start-the-event-loop call in Node.js. In Node.js the event loop is entered after executing the script that has been input. Node.js exits this loop when finally no callbacks are there to be performed. It shows similar behaviour like that of a browser JavaScript where the event loop can't be seen by the user. HTTP is a the main handler of Node.js, which helps with low latency and streaming . It makes Node.js well suited for a web library framework foundation.

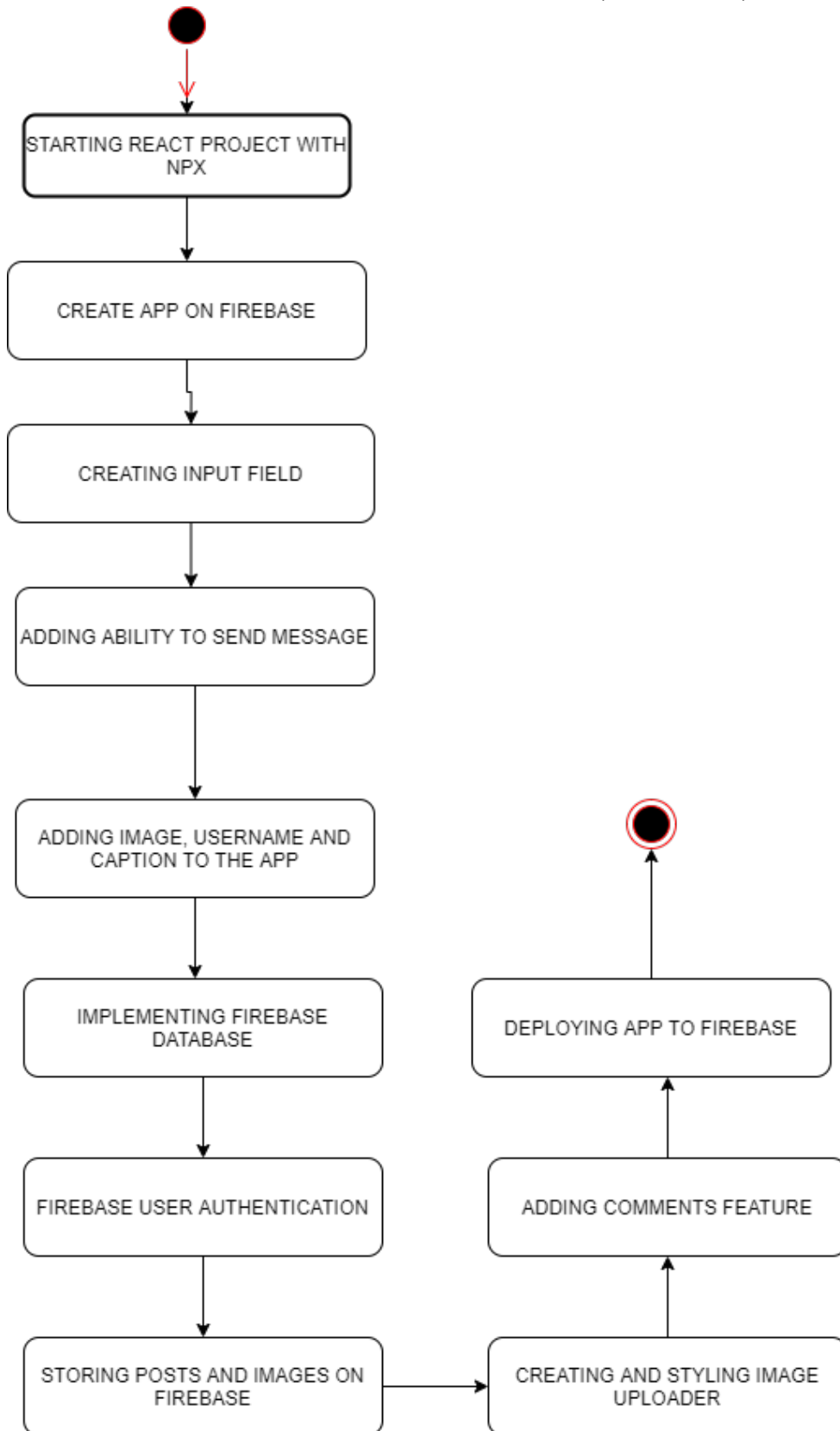
Node.js programs being designed without threads doesn't necessarily mean you can't take advantage of the multiple cores in your environment. Child processes are brought into action by using `child_process.fork()` API, these are designed to be easy to communicate with and built upon same interface is a cluster module, that allows sharing of sockets between the processes to edo load balancing over multiple cores.

3.4 Flowchart



Graph 3.1. Flow diagram depicting procedures to be followed in app

3.5 IMPLEMENTATION OF PHASES/STAGES (DIAGRAM)



CHAPTER 4

PERFORMANCE ANALYSIS

In this chapter we will be discussing the efficiency of the social media app that we are trying to build, that we researched while getting to know about the project. With the definitions that are quite basic and the initial implementation ideas about the processes to be followed, the app has been brought to life. Furthermore, we will be discussing the pros and cons that lead each of the approaches to make it suitable for achieving the desired best results. Here is a chart that we found online depicting analysis of clone detection techniques where it classifies various social media applications on the basis of approach used, parameters for detection etc and it surely helped us achieve a better understanding of different social media applications and their functioning.

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 1 st 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 4.1:- Analysis of clone detection techniques

Auth System With base of operations and React Native base of operations Authentication provides backend services & easy-to-use SDKs to certify users to your app. It supports authentication victimization passwords, phone numbers, fashionable federate identity suppliers like Google, Facebook and Twitter, and more. In most situations victimization Authentication, you'll need to understand whether or not your users square measure presently signed-in or signed-out of your application. The module provides a way referred to as `onAuthStateChanged` that permits you to purchase the users current authentication state, and receive an incident whenever that state changes it. It is vital to recollect the `onAuthStateChanged` beholder is asynchronous and can trigger associate initial state once a reference to base of operations has been established. thus it's vital to setup associate "initializing" state that blocks render of our main application while the affiliation is established it.

Save Data using firebase

Storing and retrieving user-generated knowledge is pretty common in mobile and internet apps today, and intrinsically, there square measure totally different services that provide mobile and internet developers the flexibility to store knowledge. Among these services is Google's base of operations. base of operations could be a BaaS backend-as-a-service which suggests it permits each internet and mobile developers to perform common backend tasks like user authentication and making databases with no want for maintenance or repairs.

React Native, on the opposite hand, could be a platform that enables developers to use their data of React.

Framework, to make native apps that employment across each automaton and iOS devices. React Native improves productivity as developers don't ought to manage 2 totally different codebases for each the iOS and automaton versions of a selected app.

In this article, you'll learn the way to store, retrieve, and update user-generated knowledge with base of operations. At the top of this text, we'll have designed a disorder automaton app that saves and retrieves the various disorder things in a very info, victimization info service provided by base of operations.

Handling Navigation on React Native

Mobile apps square measure seldom created of one screen. Managing the presentation of, and transition between, multiple screens is usually handled by what's called a navigator. This guide covers the assorted navigation elements obtainable in React Native. If you're obtaining started with navigation you will likely need to use React Navigation. React Navigation provides a simple navigation answer, with the flexibility to gift common stack navigation and tabbed navigation .

If you would like to attain a native look and feel on each automaton and iOS, or you are desegregation React Native into associate app that already manages navigation natively, the subsequent library provides native navigation on each platforms: react-native. The base of operations SDKs for Cloud Storage add Google security to file uploads and downloads for your base of operations apps, in spite of network quality. You can use our SDKs to store pictures, audio, video, or alternative user-generated content. On the server you'll be able to use Google Cloud Storage genus Apis to access a similar files. Displaying User Profile With base of operations Firestore , You can customise the email templet that's utilized in Authentication section of the base of operations console, on the email Templates page.

It is conjointly doable to pass state via a continue address to send back to the app once causing a countersign reset email. in addition you'll be able to localize the countersign reset email by changing the language code on the Auth instance before causing the e-mail.

Why to use React:

1. quick Learning Curve

React is extremely an easy and light-weight library that solely deals with the read layer. it's not a beast like alternative MV* frameworks like Angular or fragment. Any Javascript developer will perceive the fundamentals and begin developing associate awful internet application when solely a few of days reading tutorial.

As the React guide says 'Thinking in React' is also a bit totally different than you wont to since it brings a brand new approach to the table, however it'll become a lot of easier and natural as you gain it.

2. Reusable Components:- React provides a part based mostly structure. elements square measure your Lego items. you begin with small elements like button, checkbox, dropdown etc. and also the you produce wrapper elements composed of these smaller elements. so you write higher level wrapper elements. And, it goes on like that till you've got this one root part which part is your app.Each part decides however it ought to be rendered. every part has its own internal logic. This approach has some superb results. you'll be able to re-use elements anyplace want it.

3. quick render with Virtual DOM

When you square measure on the point of develop an online application that involves high user interaction and examine updates, just like the new type builder on JotForm four.0, you've got to contemplate the doable performance problems. though today's javascript engines square measure quick enough to handle such complicated applications, DOM manipulations square measure still not that quick.

4. Clean Abstraction

One of the powerful sides of React is that it provides an honest abstraction which suggests that it doesn't expose any complicated internals to the user.React doesn't dictate any patterns or design like MVC/MVVM, finally its solely concerning the read layer and you're absolve to style your app's design in any manner you see match this

5. Flux and revived

Flux design is introduced and maintained by Facebook and that they use it for his or her internet applications. It enhances React elements by unifacial knowledge flow. the most plan is to form actions that square measure musical organization. Any change on the store (data) will trigger a render for related components and the view is always kept in sync with data of it.

6. Great Developer Tools

Developer toolset is another important factor when you are choosing a development platform. There are two great tools you should be aware of: React Developer Tools and Redux Developer Tools. Both can be installed as Chrome extensions. React Developer Tools is great for inspecting react components within their hierarchy and also great for observing their current props and states.If you are using the Redux library

you should definitely take a look at Redux Developer Tools for Chrome. You can observe dispatched actions, current store states and watch changes on stores. You can also dispatch actions or modify stores and see changes reflected to your view instantly. Since all the updates are kept within the extension, time travel is even possible of it.

7. React Native

Learning React comes with a bonus: React Native. React is not a ‘write once run anywhere library’, as the creators says, it’s a ‘learn once write anywhere’ library. Yes, you can write native apps for Android and iOS using React Native. Although you will not be able to use the exact same code you wrote for web, you will be able to use the same methodology and the same architecture.

CODE :

1. APP.JS

```
import React, {useState,useEffect} from 'react';
import './App.css';
import Post from './Post';
import { db, auth }from './firebase';
import { makeStyles } from '@material-ui/core/styles';
import Modal from '@material-ui/core/Modal';
import { Button, Input } from '@material-ui/core';
import ImageUpload from './ImageUpload';
import InstagramEmbed from 'react-instagram-embed';

function getModalStyle() {
  const top = 50;
  const left = 50;

  return {
    top: `${top}%`,
    left: `${left}%`,
    transform: `translate(-${top}%, -${left}%)`,
  };
}

const useStyles = makeStyles((theme) => ({
  paper: {
```

```

    position: 'absolute',
    width: 400,
    backgroundColor: theme.palette.background.paper,
    border: '2px solid #000',
    boxShadow: theme.shadows[5],
    padding: theme.spacing(2, 4, 3),
  },
}));

function App() {
  const classes = useStyles();
  const [modalStyle] = useState(getModalStyle);

  const [posts, setPosts] = useState([]);
  const [open, setOpen] = useState(false);
  const [openSignIn, setOpenSignIn] = useState(false);
  const [username, setUsername] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [user, setUser] = useState(null);

  useEffect(() =>{
    const unsubscribe = auth.onAuthStateChanged((authUser)=>{
      if(authUser){
        //user has logged in
        console.log(authUser);
        setUser(authUser);
      } else {
        //user has logged out
        setUser(null);
      }
    })
  })

  return () =>{
    //perform some cleanup actions
    unsubscribe();
  }
}, [user, username]);

//useEffect runs a piece of code based on a specific conditions

```

```

useEffect(()=>{
  //this is where the code runs

db.collection('posts').orderBy('timestamp','desc').onSnapshot(snapshot =>{
  //everytime a new post is added, this code fires
  setPosts(snapshot.docs.map( doc =>({
    id:doc.id,
    post: doc.data()
  })));
})
}, []);

const signUp= (event) =>{
  event.preventDefault();

  auth.createUserWithEmailAndPassword(email, password)
  .then((authUser) => {
    return authUser.user.updateProfile({
      displayName: username
    })
  })
  .catch((error) => alert(error.message));
}

const signIn =(event) => {
  event.preventDefault();

  auth
  .signInWithEmailAndPassword(email, password)
  .catch((error) => alert(error.message))

  setOpenSignIn(false);
}

return (
  <div className="app">
    <Modal
      open={open}
      onClose={() => setOpen(false)}

```

```

    >
      <div style={modalStyle} className={classes.paper}>
        <form className="app__signup">
          <center>
            
          </center>

          <Input
            placeholder="username"
            type="text"
            value={username}
            onChange={(e) => setUsername(e.target.value)}
          />
          <Input
            placeholder="email"
            type="text"
            value={email}
            onChange={(e) => setEmail(e.target.value)}
          />
          <Input
            placeholder="password"
            type="password"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
          <Button type = "submit" onClick={signUp}>Sign
up</Button>
        </form>

      </div>
    </Modal>

    <Modal
      open={openSignIn}

```



```

    onClose={() => setOpenSignIn(false)}

  >
    <div style={modalStyle} className={classes.paper}>
      <form className="app__signup">
        <center>
          
        </center>

        <Input
          placeholder="email"
          type="text"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
        <Input
          placeholder="password"
          type="password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
        />
        <Button type = "submit" onClick={signIn}>Sign
In</Button>
      </form>

    </div>
  </Modal>

  <div className="app__header">
    
    {user ? (
      <Button onClick={() => auth.signOut()}>Logout</Button>
    ) :(
      <div className="app__loginContainer">
        <Button onClick={() => setOpenSignIn(true)}>Sign In</Button>
        <Button onClick={() => setOpen(true)}>Sign Up</Button>
      </div>
    )}
  </div>
  <div className="app__posts">
    <div className="app__postsLeft">
      {
        posts.map(({id, post}) =>(
          <Post key={id} postId={id} username={post.username}
caption={post.caption} imageUrl={post.imageUrl}/>
        ))
      }
    </div>

    <div className="app__postsRight">
      <InstagramEmbed
        url='https://instagr.am/p/Zw9o4/'
        //clientAccessToken='123|456'
        maxWidth={320}
        hideCaption={false}
        containerTagName='div'
        protocol=''
        injectScript
        onLoading={() => {}}
        onSuccess={() => {}}
        onAfterRender={() => {}}
        onFailure={() => {}}
      />
    </div>
  </div>

  {user?.displayName ? (
    <ImageUpload username={user.displayName} />

```

```

    ): (
      <h3>Sorry !! You must Login to Upload</h3>
    )}
  </div>

);
}

export default App;

```

Majority of code is not present here due to obvious copying reasons and this is the redacted part of code.

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;
  }
```

```
.post_text {
  font-weight: normal;
  padding: 20px;
}
```

```
.post_header {
  display: flex;
  align-items: center;
  padding: 20px;
}
```

```
.post_avatar {
  margin-right: 10px;
}
```

Firestore:

```
import firebase from "firebase";
const firebaseApp = firebase.initializeApp({
  apiKey: "*****",
  authDomain: "*****.firebaseapp.com",
  databaseURL: "https://Socialmedia-clone-react-a3857-default-
rtdb.firebaseio.com",
  projectId: "*****",

  storageBucket: "socialmedia-clone-react-a3857.appspot.com",
  messagingSenderId: "9041****",
  appId: "1:90417****:web:38d10081*****",
  measurementId: "G-PZ*****"
```

```
});  
const db = firebaseApp.firestore();  
const auth = firebaseApp.auth();  
const storage = firebaseApp.storage();  
  
export { db, auth, storage };
```

Outputs:

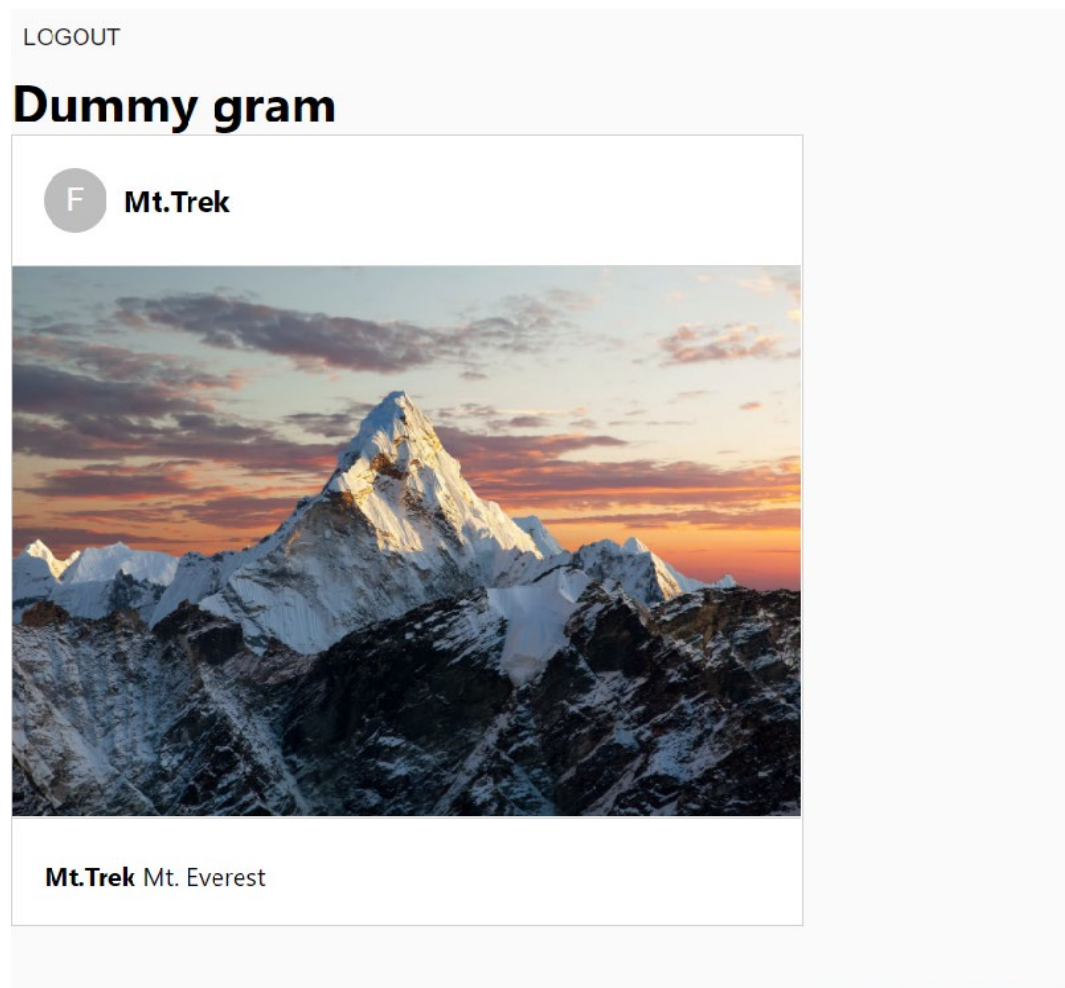


Figure:- 4.1 UI of Social Media app

The web application has started taking the form of a similar look to an authentic Social media app.

Alot of functionalities are implemented. We have successfully created a login/signup option and you can surf on the feed page.

Enter a caption... Choose File No file chosen UPLOAD

FIGURE 4.2:- Image Uploader

Post can be uploaded by the user who is registered with the image, caption and username inserted from the firebase.

We have also created a upload feature which choose an image you want to upload from your local device and upload it on the application.

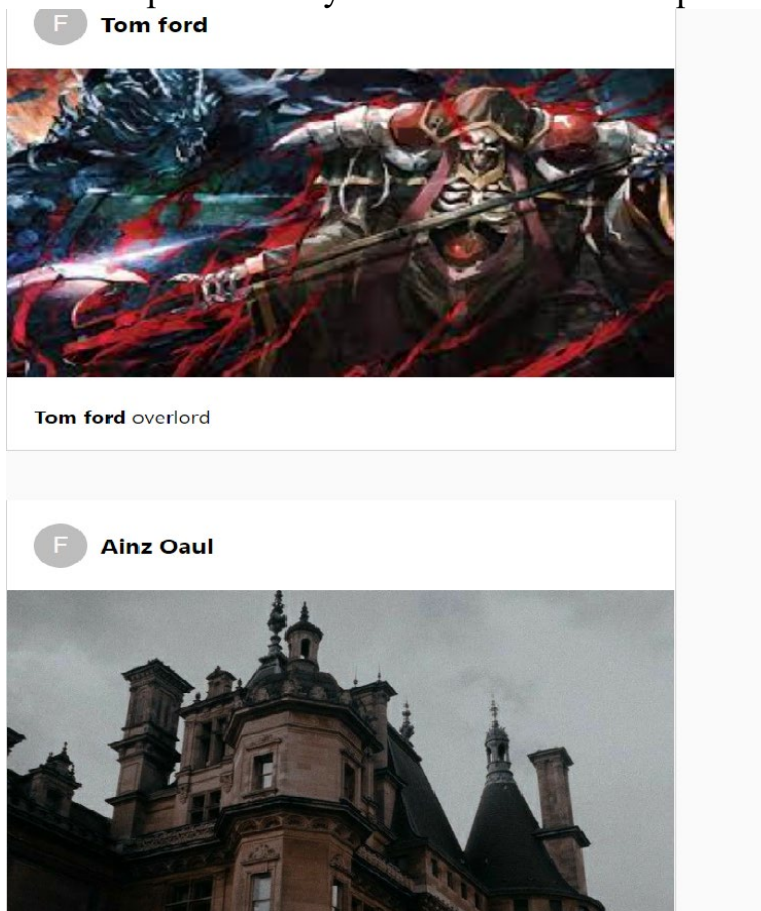


Figure 4.3:- UI

The users can post the caption of their choice and later on they will be able to comment on each other's post.

The UI has been designed with the help of react js and it has been our main objective to make the User Interface and experience feel like they can easily navigate through the application.

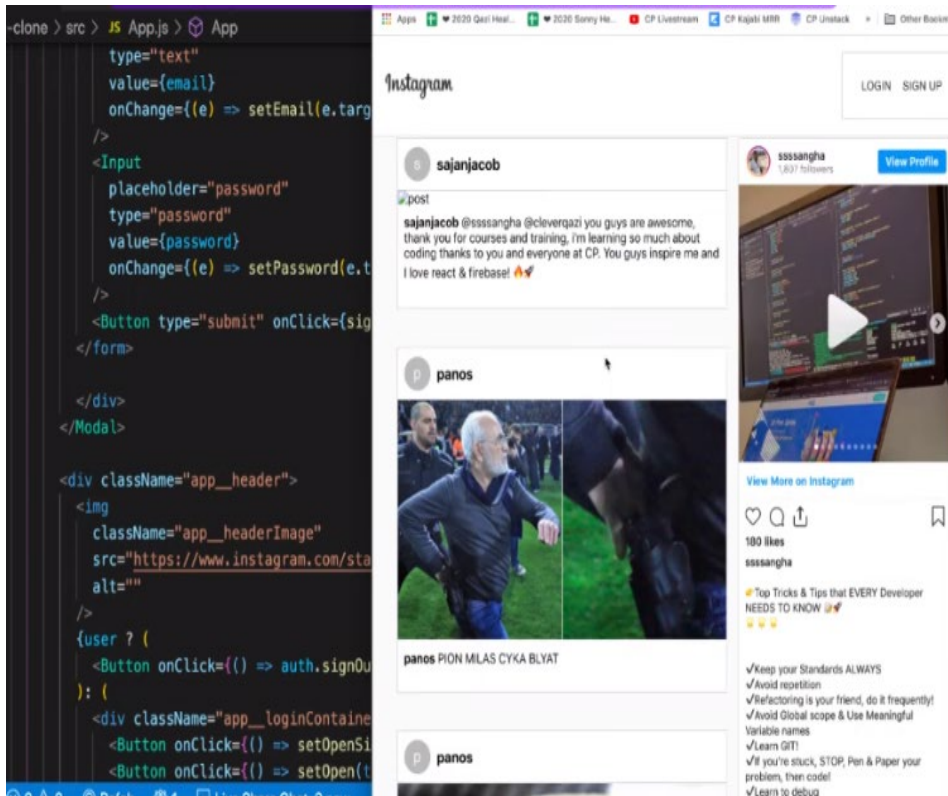


Figure 4.4: finalized look of app.

CHAPTER 5 CONCLUSIONS

We have included 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.

The main reason for using React is to be very fast, scalable and simple. We developed the social media app using react having the features like fast, accurate, scalable and simple.

In react 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 :”Nexttone 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)
- [6] :D. Muriyatmoko,N A S Asy’ari, M S Arif: “Android Radio Streaming Apps for Songgolangit FM Ponorogo”:
The 1st International Conference on Engineering and Applied Science 21 August 2019, Madiun, Indonesia: (2019)
- [7] : Analysis of social networking sites: A study on effective communication strategy in the developing of brand communication P. Sri Jothi*, M. Neelamalar and R. Shakthi Prasad.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 24-06-2021

Type of Document (Tick):

B.Tech Project Report

Name: FALAQ SHAHRAJ KHAN and Shubhra Pundir Department: CSE Enrolment No 171354

, 171262 Contact No. 7018262743

E-mail.FALAQMADDY@GMAIL.COM, 171262@juitsolan.in

Name of the Supervisor: DR. KAPIL SHARMA

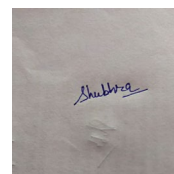
Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): SOCIAL MEDIA APP USING REACT.JS

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =41
- Total No. of Preliminary pages =3
- Total No. of pages accommodate bibliography/references =1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 28 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ORIGINALITY REPORT

28%

SIMILARITY INDEX

21%

INTERNET SOURCES

9%

PUBLICATIONS

19%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Asian Institute of Technology Student Paper	7%
2	towardsdatascience.com Internet Source	6%
3	Submitted to Jaypee University of Information Technology Student Paper	5%
4	github.com Internet Source	3%
5	Satish Kumar Satti, K. Suganya Devi, Prasenjit Dhar, P. Srinivasan. "A machine learning approach for detecting and tracking road boundary lanes", ICT Express, 2020 Publication	1%
6	www.thushv.com Internet Source	1%
7	www.coursehero.com Internet Source	1%
